UNIVERSITY OF CYPRUS

DEPARTMENT OF COMPUTER SCIENCE

PROSPECTUS

Academic Year 2022-2023

POSTAL ADDRESS
Department of
Computer Science
University of Cyprus
P.O. Box 20537
CY-1678 Nicosia
CYPRUS

Tel.: +357-22892700
Fax: +357-22892701
E-mail: cs@ucy.ac.cy
Web: http://www.cs.ucy.ac.cy
Twitter: @csdeptucy

Open Letter from the Chairperson of the Department

Dear Students,

This prospectus contains a concise and comprehensive presentation of the Computer Science Department of the University of Cyprus. Furthermore, in this prospectus you will find information regarding your studies at the Department, including undergraduate and graduate curricula, course descriptions, rules of attendance, prerequisites for graduation, short biographies of the faculty, etc. Up to date information, announcements, news, etc. are published through the website of the department at http://www.cs.ucy.ac.cy and on our Twitter (@csdeptucy) or Facebook accounts; also through e-mail and on the announcement boards near the Secretariat of the Department. For personal advice on your studies, or for scientific/ professional orientation, or for any other matter that concerns you and may affect your studies, I strongly encourage you to keep regular contact with the academic advisor assigned to you during your enrolment to the Department.

The Computer Science discipline, albeit a relatively young scientific field, has already been recognized as a key scientific and technological discipline with impressive development and a strong influence on other sciences and the society at large. The growing significance and broad applicability of Computer Science has established the Department of Computer Science as one of the most dynamic and competitive departments in the University of Cyprus. The skills and high educational level that our students gain during their studies are acknowledged both by the local IT industry as well as abroad, as our graduates achieve commendable performance either as employees in the domestic industry or as postgraduate students at leading Universities in Europe and America. The annual student intake of the Department of Computer Science consists of 80 undergraduate students and approximately 35-40 graduate students. Currently, around 450 students are enrolled in the Department, of whom 82% are undergraduate students and the rest are postgraduate students of Masters level or doctoral students. Our Department is working systematically towards attracting outstanding students by organizing and supporting annual Computer Science seminars and competitions, such as the Information Day for High School Students and the Logipaignion game-creation contest. In fact, over the last five years (2017-2021) during the Pancyprian Examinations, the Department was ranked first among all programmes of study in pure and applied sciences, and engineering, regarding the number of candidates having it as their top preference. These are particularly strong pieces of evidence in favour of the overall standing of the Department.

Both the undergraduate and postgraduate curricula have been designed to meet the current challenges and advancements in Computer Science, and cover theoretical foundations, technical knowledge, and

experimental methodologies of Computer Science with a strong problem-solving focus. The Department follows on a continuous basis and contributes itself towards the scientific developments and advances of Computer Science, revising and modernizing at regular intervals its curricula, enriching its catalogue of advanced electives, and enhancing the interdisciplinary content of its education provision. In addition, the Department offers a minor in Computer Science for students of other departments of the University of Cyprus. More details are included in this prospectus.

At the postgraduate level, our department offers a Ph.D. program in Computer Science and a number of Masters programmes, including both programmes with a research orientation and programmes with a professional orientation. In addition, from the academic year 2017/18, the Department offers in English a joint Masters programme in "Cognitive Systems", by distance learning, in collaboration with the Open University of Cyprus and the Department of Psychology of the University of Cyprus. Last academic year a new interdisciplinary and inter-departmental Masters programme in Data Science, taught in English, in collaboration with the Departments of Mathematics and Statistics, and Public and Business Administration started its delivery. Yet another specialized Masters programme in Artificial Intelligence, also taught in English, has been developed with financial support from the European Commission and has started its delivery this academic year.

All programmes of the Department comply with the regulations of the European Credit Transfer and Accumulation System (ECTS). Every student, on the completion of his/her studies receives a Diploma Supplement together with his/her University degree. During your studies, you are strongly encouraged to participate in the European programme ERASMUS+ that supports student exchanges, thus allowing you to spend a semester of your studies at another academic institute within the European Higher Education Area.

Our Department is in constant dialogue with industry, relevant government departments and centers of excellence abroad to contribute decisively in promoting Cyprus as a center of services, expertise and innovation in Computer, Information and Communication technologies. Also, the Department is working vigorously for the international promotion of our students' skills through attendance and/or awards in international competitions in programming, innovation, entrepreneurship, etc. Overall, our goal is to bestow our students with knowledge, skills and competencies that would enable them to push the frontiers of the Cypriot IT industry towards new highly competitive areas in comparison with the corresponding industries of other countries in our geographical region as well as in Europe.

In these efforts the strong presence of the Department in research is vital. Since its foundation, over twenty-seven years ago, the Department has developed highly important activities and contributes in the promotion of science and knowledge worldwide. Particularly important is the participation of the Department in research programs of the European Union (EU), as well as domestic programs of the Research and Innovation Foundation (RIF) of Cyprus. Specifically, since 2007, faculty members of the Department have attracted more than 160 research projects from the EU and the RIF with a total budget exceeding €27 mil. Most of this funding has been used to subsidize hundreds of young scientists - researchers, doctoral students, postdoctoral associates and graduate students from Cyprus and abroad, to develop modern research and teaching infrastructure and the development of innovative software systems, hardware and applications. Research results are presented by faculty members and researchers of the Department each year in top Computer Science journals and conferences and are utilized by the international scientific community and industry. In addition, a particularly valuable contribution of the Department is its support towards governmental actions for the uptake of Information and Communication Technologies by the public sector, the promotion of the Internet and the Web in Cypriot businesses and society, and the technological modernization of Secondary

Education and Health Services.

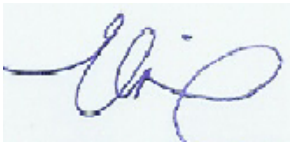
During your studies, I urge you to actively engage in the Departmental teaching and research activities to get the utmost regarding knowledge and experience that the Department can offer to you. Also, I invite you to take advantage of courses in other disciplines offered by the various Departments of the University, in order to enhance your knowledge and gain a more comprehensive and universal education. This opportunity is also available through the European Student Exchange Programme ERASMUS+ mentioned above.

Dear students,

The unprecedented situation we have experienced, and still experiencing, due to the new Covid-19 virus, in addition to the important lessons it has given us, demonstrating in practice that with collectivism such crises can be overcome and even turned into opportunities, highlighted the importance of technology and of information tools. Without them, it would not be possible for us to continue to operate and complete the last three academic years smoothly.

On behalf of all the members of the Department, I once again welcome you and wish everyone a fruitful, creative and productive year!

With best wishes,



Professor Elpida Keravnou-Papailiou,

Chairperson of the Department of Computer Science

TABLE OF CONTENTS

STAFF

Academic Faculty

Keravnou-Papailiou Elpida
(Chairperson)
Chris Christodoulou *(Vice-Chairperson)*
Elias Athanasopoulos
Andreas Aristidou
Yiorgos Chrysanthou
Marios D.Dikaiakos
Yannis Dimopoulos
Chryssis Georgiou
Antonis Kakas
Georgia Kapitsaki
Marios Mavronicolas
George Pallis
George A Papadopoulos
Constantinos Pattichis
Andreas Pitsillides
Andreas Pieris
Yanos Sazeides
Anna Philippou
Vasos Vassiliou
Haris Volos
Demetris Zeinalipour

Visiting Academics

to be updated

Special Teaching Scientists

Zinonas Antoniou
Christoforos Christoforou
Marios Neofytou
Maria Papaioannou
Irene Schiza
Marios Belk

Special Teaching Staff

Pavlos Antoniou
Pyrros Bratskas
George Hadjipollas
Yiannakis Mylonas
Christoforos Panagiotou
Petros Panayi

Secretariat

Dora Georgiou
Savvoula Efstathiou
Maria Kittira
Melina Menelaou-Chrysostomou

Administrative Support to Research Programmes

Helen Kakos

Information Systems Technical Support

Andreas Kasenides
Andry Michaelidou
Maria Tsiolakki
Andreas Filippou

Academic Faculty Information

Academic Faculty

Elias Athanasopoulos
Associate Professor
Office: FST 01 B105
Phone: +357-22892754
Email: eliasathan@ucy.ac.cy

Andreas Aristidou
Assistant Professor
Office: FST 01 B113
Phone: +357- 22892698
Email: andarist@ucy.ac.cy

Chris Christodoulou
Professor
Office: FST 01 113
Phone: +357-22892752
Email: cchrist@ucy.ac.cy

Yiorgos Chrysanthou
Professor
Office: FST 01 013
Phone: +357-22892719
Email: yiorgos@ucy.ac.cy

Marios D. Dikaiakos
Professor
Office: FST 01 012
Phone: +357 22892720
Email: mdd@ucy.ac.cy

Yannis Dimopoulos
Professor
Office: FST 01 014
Phone: +357-22892718
Email: yannis@ucy.ac.cy

Chryssis Georgiou
Professor
Office: FST 01 115
Office: FST 01 010
Email: chryssis@ucy.ac.cy

Antonis Kakas
Professor
Office: FST 01 110
Phone: +357-22892706
Email: antonis@ucy.ac.cy

Georgia Kapitsaki
Associate Professor
Office: FST 01 119
Phone: +357-22892692
Email: gkapi@ucy.ac.cy

Elpida Keravnou-Papailiou
Professor
Office: FST 01 117
Phone: +357-22892694
Email: elpida@ucy.ac.cy

Marios Mavronicolas
Professor
Office: FST 01 106
Phone: +357-22892702
Email: mavronic@ucy.ac.cy

George Pallis
Associate Professor
Office: FST 01 B119
Phone: +357-22892743
Email: gpallis@ucy.ac.cy

George Angelos Papadopoulos
Professor
Office: FST 01 118
Phone: +357-22892693
Email: george@ucy.ac.cy

Constantinos Pattichis
Professor
Office: FST 01 114
Phone: +357-22892697
Email: pattichi@ucy.ac.cy

Anna Philippou
Professor
Office: FST 01 105
Phone: +357-22892699
Email: annap@ucy.ac.cy

Andreas Pieris
Assistant Professor
Office: FST 01 111
Phone: +357-22892705
Email: pieris.andreas@ucy.ac.cy

Andreas Pitsillides
Professor
Office: FST 01 116
Phone: +357-22892695
Email: andreas.pitsillides@ucy.ac.cy

Yanos Sazeides
Associate Professor
Office: FST 01 109
Phone: +357-22892704
Email: yanos@ucy.ac.cy

Vasos Vassiliou
Associate Professor
Office: FST 01 111
Office: FST 01 B114
Email: vasosv@ucy.ac.cy

Haris Volos
Assistant Professor
Office: FST 01 115
Phone: +357-22892696
Email: hvolos01@ucy.ac.cy

Demetris Zeinalipour
Associate Professor
Office: FST 01 B106
Phone: +357-22892755
Email: dzeina@ucy.ac.cy

Visting Faculty

to be updated

Special Teaching Scientists

to be updated

Special Teaching Staff

Pavlos Antoniou
Office: FST 01 B109
Phone: +357-22893927
Email: antoniou.pavlos@ucy.ac.cy

Pyrros Bratskas
Office: FST 01 B118
Phone: +357-22893930
Email: bratskas@ucy.ac.cy

Giorgos Hadjipollas
Office: FST 01 B110
Phone: +357-22893932
Email: hpollas@ucy.ac.cy

Yiannakis Mylonas
Office: FST 01 B115
Phone: +357-22893931
Email: mylonasy@ucy.ac.cy

Petros Panagi
Office: FST 01 B118
Phone: +357-22893926
Email: petrosp@ucy.ac.cy

Christophoros Panagiotou
Office: FST 02 B104
Phone: +357-22893928
Email: panchris@ucy.ac.cy

Information Systems Technical Support

Andreas Filippou
Assistant IT Officer
Office: FST 01 001
Phone: +357-22892711
Email: andreasf@ucy.ac.cy

Andreas Kasenides
Senior IT Officer
Office: FST 01 018
Phone: +357-22892714
Email: ank@ucy.ac.cy

Andry Michaelidou
IT Officer
Office: FST 01 B117
Phone: +357-22892734
Email: andrim@ucy.ac.cy

Maria Tsiolakki
Senior Assistant IT Officer
Office: FST 01 B117
Phone: +357-22892727
Email: tmaria@ucy.ac.cy

Administrative Support to Research Programmes

Helen Kakos
University Assistant Officer
Office: FST 01 017
Phone: +357-22892713
Email: kakos@ucy.ac.cy

Secretariat

Melina Menelaou Chrisostomou
Secretary
Office: FST 01 015
Phone: +357-22892721
Email: melina@ucy.ac.cy

Savvoula Efstathiou
Secretary
Office: FST 01 009
Phone: +357-22892669
Email: savvoula@ucy.ac.cy

Dora Georgiou
Secretary
Office: FST 01 019
Phone: +357-22892722
Email: addora@ucy.ac.cy

Maria Kittira
Secretary
Office: FST 01 007
Phone: +357-22892700
Email: manak@ucy.ac.cy

ACADEMIC CALENDAR **2022-2023**

	Fall Semester 2022-2023	**Spring Semester 2022-2023**
Registration and orientation of new entering students	01/09/2022 – 02/09/2022	11/01/2023– 12/01/2023
Classes begin	05/09/2022	16/01/2023
Deadline for course selection	09/09/2022	20/01/2023
Deadline for dropping a course	23/09/2022	03/02/2023
Deadline for withdrawal	21/10/2022	03/03/2023
Easter Holidays		10/04/2023 - 23/04/2023
End of classes	02/12/2022	28/04/2023
Examination Period	09/12/2022 - 23/12/2022	12/05/2023 - 26/05/2023
Christmas Holidays	24/12/2022 – 10/01/2023	
Public Holidays	1st October 28th October 6th January	27th February 25th March 1st April 16th April (Easter) 1st May 5th June

Introduction

The Computer Science field at its early days of birth had a specific scope, mainly concerned with the automation of mathematical calculations; quickly though, it developed into an exciting amalgam of science, technology, and theory. Today, Computer Science addresses a wide range of topics such as extending the range of problems which can be effectively solved using computers; creation, maintenance, and optimization of software and hardware for high performance computer systems; the way in which humans reason, discuss, and plans their activities; modeling the operation of the human brain; and the role of language and logic in problem-solving. Consequently, Computer Science has direct links with all of the Natural Sciences, and with many other branches of human knowledge such as Philosophy, Psychology, Cognitive Sciences, Linguistics, and Management. This multidisciplinary nature of our relatively young field inherits unlimited opportunities for discovering new applications and exploring new research directions.

Nowadays, Computer Science influences essentially every field of human endeavor, with important applications in industry, commerce, economics, education, and medicine. The fast technological growth and the wide spread of computers in our lives has significantly improved the quality of our lives by offering a huge variety of new services such as performing tasks which are too dangerous or too complicated for humans and contributing to the dissemination of knowledge and expertise via knowledge-based systems. Computer networks and distributed processing techniques are widely used and have become essential in today's working environment. The initial misapprehension that people would be replaced by computers has now been dispelled by the realization that computers merely amplify people's natural abilities and provide assistance in performing their work more effectively.

The ECTS System

As of the academic year 2005-2006, the University of Cyprus has adopted the European Community Course Credit Transfer and Accumulation System, or ECTS for short.

What is ECTS?

ECTS, or *European Credit Transfer and Accumulation Credits*, was developed by the Commission of the European Communities in order to provide common procedures to guarantee academic recognition of studies uniformly throughout Europe. It provides a way of measuring and comparing learning achievements and transferring them from one European institution to another.

ECTS provides transparency through the following means:

- *ECTS Credits* which are a numerical value allocated to course units to describe the student *workload* required to complete them.
- The *Information Package*, which supplies written information to students and staff on institutions, departments/faculties, the organizations and structure of studies and course units.
- The *Diploma Supplement*, which shows students' learning achievements in a way which is comprehensive, commonly understood and easily transferable from one institution to another.

The ECTS Credits

ECTS credits are allocated units to describe the student *workload* required to complete them. They reflect the *quantity* of work each course requires *in relation* to the total quantity of work required to complete a full year of academic study at the institution that is, lectures, practical work, seminars, private work – in library or at home – and examinations or other assessment activities. ECTS credits express a *relative value*. In ECTS, *60 credits* represent the workload of a year of study and *30 credits* are given for a semester.

Credits are awarded only, when the course has been completed and all required examinations have been successfully taken.

The expected time for the completion of the Undergraduate Program of Studies is eight semesters. This time can be extended up to twelve semesters. A student is required to complete 240 ECTS units in order to acquire a Batchelor degree in Computer Science at the Undergraduate Programme of Studies. Furthermore, for a Master Degree, successful completion of at least 90 ECTS is required.

General Information

Class Attendance and Teaching

Teaching is carried out in the internationally established way through lectures, recitiation (tutorials), laboratories, etc. Students are expected to consistently participate in all activities of classes they are attending (e.g., lectures, tutorials, laboratories, etc.). The Department reserves the right to disallow students who systematically skip the activities of attended classes from taking the corresponding final examinations.

In most classes, homework is assigned regularly to help students master the course material and develop practical skills. Homework is carried out either individually or collaboratively small student groups.

Evaluation is usually based on assigned homework, written and oral examinations, etc. Special efforts are made to use continuous evaluation whenever possible.

Students are informed about the specific procedure of teaching and evaluation for each class from its class Instructor. The relevant information is included in the "General Information" sheet for the class, which is handed out to the students during the first week of classes each semester. The Instructor is available to the students during predetermined weekly office hours.

The general rules of studies of the University, the rules that govern the rights and responsibilities of the students, as well as the procedures that need to be followed, are included in separate leaflets. These are made available to the students by the Administrative Service of Academic Affairs and Student Welfare. Student must know all the rules that concern them.

Academic Advisor

A member of the Department's academic staff is assigned to each student as an Academic Advisor. The Academic Advisor provides assistance or advice on academic matters (e.g., difficulties in following a class, appropriate combinations of restricted choice classes, etc.), but also advice on other matters such as personal problems.

The role of the Academic Advisor can not be effective without the cooperation of the student. So, both undergraduate and graduate students, especially the freshmen, are strongly encouraged to have frequently meetings with their academic advisors, in order to achieve better preparation and organization of their program of study as well as to resolve related issues and problems. It should be noted that the students are held responsible for their final decisions regarding their academic choices. However, they should keep informed their Academic Advisor of their decisions.

Student Representation

Six elected representatives of the students are currently participating in the Departmental Board as members. Furthermore, during each Fall Semester, the Department requests that students elect two representatives per year in order to facilitate intercommunication between the year's students and the Department.

Secretariat

Daily activities can be carried out through the Departmental Secretariat. Students are informed by their Academic Advisors on services or other facilities the Secretariat can provide as well as about the specific timings the Secretariat is available to the Students. In particular, the Secretariat can provide any kind of general information regarding the Department or the University.

Course Schedule

The Class Schedule is not included in this Prospectus, but it is handed out to the students at the beginning of each Semester. Although every possible effort is made to satisfy the scheduling of any reasonable combination of classes, there is, however, a possibility that some students will face difficulties in combining certain of their (restricted or unrestricted) choices due to, e.g., concurrent class meetings. In such cases, students are urged to immediately notify their Academic Advisors.

Library Use

The University's Library is equipped with a large number of books and scientific journals in Computer Science. These include books that will assist the students in mastering the taught course materials, and, in particular, the books recommended or suggested by Instructors. Therefore, the students should have their student ID in order to have access to the library facilities. Regulations for using the Library are separately handed out to the students at the beginning of each academic year.

Electronic Notices / Departmental Website

The Department uses electronic mail for the fast dissemination of information among the students and its Faculty/Staff members.

Also, all necessary information (courses, talks, announcements, etc) can be found at the Departmental website (https://www.cs.ucy.ac.cy). There is also a spectial section for students called Student's Corner where students can obtain up-to-date important academic information. Furthermore, the department is using social media (facebook and twiter pages) for the better communication among its members.

Electronic Mail

The use of Electronic Mail for communication between academic staff and students, and also between students themselves, is considered a must, and students are strongly encouraged to learn as soon as possible how to effectively use Electronic Mail. However, it is emphasized that the use of Electronic Mail is not a right of the students, but rather a service provided by the Department. In cases where inappropriate use of Electronic Mail is noticed, the Department reserves the right to disallow its use.

Laboratory Equipment

A significant part of our Curricula is of practical content. The Department has five laboratories, fully equipped for teaching and research in Computer Science.

- The *Postgraduate Laboratory (201)* consists of 30 last generation computers with Microsoft Windows 7 Enterprise operating system and special software for the support of specialized courses at the graduate level and, more specifically, in the directions of Internet Technologies and Intelligent Systems.
- The *General Teaching Unix Laboratories (B103)* and (103) consists of 33 workstations of the latest generation equipped with the Linux (CentOS 6.0) operating system and Free NX server, and they provide software for the development of various applications.
- The *General Undergraduate Laboratory 1 (B121)* consists of 30 workstations of the latest generation, equipped with the operating system Windows 7, and provided with

software for the development of various applications.

- The *General Undergraduate Laboratory 2 (B123)* consists of 31 workstations each, of the latest generation, equipped with the operating system Windows 7, and it provides software for the development of various applications.
- The *Digital Systems Design and Microprocessor Laboratory (101)* consists of 27 last generation computers with Microsoft Windows 7 Enterprise operating system. Furthermore, there are digital logic and microprocessor boards, oscilloscopes, digital multimeters, logic probes, signal generators and integrated circuit testers. This equipment is used for practical training in hardware. In addition, this Laboratory includes peripheral devices for training in the design of microprocessor systems, and in the development of prototypes.
- The *Walk-in Laboratory (B101),* in which both personnel and students may bring their portable computers and connect to the Internet and the departmental computer systems.
- The *Tele-education room (148)* is a lecture room of 20 seats and it can be used for: (i) data distribution to remote users, (ii) live and on demand streaming of lectures and presentations, (iii) lecture capturing on DVDs.

More information on the equipment is provided on the website of the Information Systems Technical Support (http://its.cs.ucy.ac.cy/).

Laboratory Equipment Use Regulations

Students are kindly requested to respect fundamental principles of professional behavior regarding health and safety in common rooms and responsible use of laboratory equipment. In particular, the following are indicative examples of disallowed tasks:

- Access to computer systems using somebody else's account
- Inappropriate use of Electronic Mail
- Use of computer systems for purposes other than the "normal" (e.g., development of commercial products, disturbance of users, etc.)
- Use of software products other than those provided by the Computer Center, without the consent of the Computer Center or the Department, respectively.
- Attempt to access confidential information
- Copying software products that belong to others, in violation of international laws of liability.

Specific information regarding Regulations and Time Table for using the Computer Laboratories are handed out to the students at the beginning of each academic year.

Awards

For each academic year, the Department grants, on the basis of academic merit, awards to its undergraduate students. These awards have been founded by external sponsors. In the academic year 2020/2021, the following were awarded:

[1] *«Award MetaQuotes Software»* from MetaQuotes Software to the graduating undergraduate and master students with excellent academic performance (Monetary award of 20000€)

[1] *«Award Logicom (Public)»* from Logicom Public Ltd to the graduating student who has attained the highest overall academic performance (Monetary award of 1000€)

[2] *«Award JCC»* in the memory of the deceased General Director of JCC Taki Fekkou to the graduating student who has attained the second highest overall academic performance (Monetary award of 500€)

[3] *«Award HF Markets»* from HF Markets (Europe) Ltd to the graduating student who has attained the second highest overall academic performance. (Monetary award of 500€)

[4] *«Award SEPAIK»* from SEPAIK Union to the graduating student who has attained the highest overall academic performance. (Monetary award of 350€)

Also, the Department itself grants the following awards to its undergraduate students:

[1] *Special Award* to the undergraduate student who is completing the Minor Program of Studies in Computer Science and has attained the highest overall academic performance in the courses of the minor program. (Monetary award of 340€)

[2] *Special Award* for the graduating student who has demonstrated exceptional social contribution. (Monetary award of 340€)

[3] *Award* for the graduate student who has demonstrated outstanding and praise-deserving effort in Memory Polyvios Polyviou. (Monetary award of 500€)

[4] *Excellence Award in the memory of the deceased Professor George Samaras* to the graduating student who has attained in combination the highest academic performance in the compulsory course Database Systems and the highest overall academic performance. (Monetary award of 500€)

[5] *Department Award of excellence* to the graduating students of the Department of Computer Science who have achieved an excellent performance . (Over 8.5/10) (Monetary award of 200€/each)

Also, the Department grants, every year, the following awards to its postgraduate students:

[1] *«1st Award of master programme in Computer Science»* for the graduating student who has attained the highest overall academic performance. (Monetary award of 680€)

[2] *«2nd Award of master programme in Computer Science»* for the graduating student who has attained the second highest overall academic performance. (Monetary award of 340€)

[3] *«Award of Mariou Tsakalaki»*, donated by the family of the deceased Marios Tsakalakis (who served as System Architect in the former Laiki Banj), for the graduating student who has attained the highest overall academic performance. (Monetary award of 500€)

The Department awards a certificate of excellence to the student with the best overall academic performance in her year for the 1st, 2nd and 3rd academic year, as well as certificates to all students that have been distinguished in national and international competitions.

** Awards may differ each academic year following the guidelines of the sponsors.

Administrative Duties of Academic Staff

Committees/Groups

[1] **Postgraduate Program Committee:** *V. Vasileiou (coordinator), Y. Dimopoulos, Y. Sazeides, Chr. Christodoulou*

[2] **Undergraduate Program Committee:** *A. Philippou (coordinator), C. Pattichis, A. Pieris*

[3] **Undergraduate Transfers Committee and Correspondence Students:** *G. Papadopoulos (coordinator), Y. Chrysanthou, Y. Dimopoulos*

[4] **Erasmus+ Program:** *G. Kapitsaki (coordinator), Y. Dimopoulos, A. Philippou*

[5] **Communication and Promotion:** *G. Kapitsaki (publication coordinator), G. Pallis (website coordinator)*

[6] **Logipaignion Contest:** *D. Zeinalipour, A. Aristidou*

[7] **Hmerida Pliroforikis:** *rotating each year*

[8] **Strategic Planning:** *E. Keravnou-Papailiou (coordinator), A. Philippou, Chr. Christodoulou, M. Dikaiakos*

[9] **Liaison with the Industry and Student Internships:** *E. Athanasopoulos (coordinator), M. Dikaiakos, Chr. Georgiou, H.Volos*

[10] **Liaison with ERCIM:** *A. Aristidou (contact point), G. Kapitsaki (editorial board)*

[11] **Master in Cognitive Systems**: *A. Kakas*

[12] **Summer School**: *V. Vasileiou, Y. Chrysanthou*

[13] **Conact Point for Computer Science Student Club**: *Y. Sazeides*

[14] **Computer Systems and Infrastructures Committee**: *D. Zeinalipour (coordinator), E. Athanasopoulos, G. Pallis, H. Volos*

Representatives/Coordinators:

[1] **Library:** *A. Philippou*

[2] **Diploma Project:** *M. Mavronicolas*

[3] **Prospectus:** *A. Pieris*

[4] **Departmental Seminars:** *D. Zeinalipour*

[5] **Allocation of Teaching Assistantships:** *A. Pieris, G. Hadjipollas*

[6] **ACM ICPC Cyprus Competition:** *Chr. Georgiou*

[7] **International Relations:** *A. Aristidou*

[8] **YUFE (Young Universities for the Future of Europe):** *G. Kapitsaki*

[9] **Liaison with Secondary Education:** *Chr. Georgiou*

[10] **Academic advisor for Department for Students with Disabilities:** Y. Demopoulos

Undergraduate Program of Studies

According to the ECTS System, at least 240 ECTS units are required for acquiring a Batchelor's degree in Computer Science.

Objectives and Prospects

The Undergraduate Program of Studies leads to a Batchelor's degree in Computer Science. The Department aspires to prepare graduates able to pursue careers in positions of responsibility in either academia or industry, where they will effectively drive the development and application of new methods and ideas. The Department attaches particular importance to maintaining links and continuous dialogue with local industry, and we expect our graduates to play a significant role in this respect.

Any curriculum, no matter what its subject, must offer the student education in the widest sense of the term, and cultivate the desire for continuous learning, which, in turn, leads to maturity and develops the facilities for independent and critical thinking. This general objective coexists with and enriches the more specific objectives of individual programs.

We hope that our graduates will acquire a deep understanding of Computer Science, both as a science, and in terms of its more general applications and effects on society. The Curriculum covers the essential practical techniques, together with the deeper principles which underpin them. Our graduates will be well qualified to obtain immediate employment in Cyprus or abroad as IT professionals, or as teachers of Computer Science in secondary education; alternatively, they may choose to continue their studies to postgraduate level, and further pursue a career in either academia or research. Whichever career path they choose, their studies will have given them the necessary grounding to keep abreast of the incredibly rapid scientific and technological developments in Computer Science.

The expected time for the completion of the Undergraduate Program of Studies is eight semesters. This time can be extended up to twelve semesters.

Course Areas

The Department's undergraduate course material is conceptually divided into the following areas or components: *Theory, Computer Systems, Problem Solving* and *Applications.*

The *Theory* component is concerned with the foundations of Computer Science: theory and models of computation, and the design and analysis of algorithms. Generally this component aims at cultivating a formal approach to thinking, and organizing and processing information. Logic, and its role as the calculus of Computer Science, is an important topic. Essential concepts of Discrete Mathematics are taught as an integral part of related courses offered by the Department. Students are also required to follow a number of courses offered by the Department of Mathematics and Statistics in order to develop their abilities for abstraction and formal thinking, and to acquire other useful mathematical skills.

The *Computer Systems* component is concerned with hardware and software systems and elaborates on the concepts of parallel and embedded systems. It includes basic principles of computer architecture and organization, operating systems, programming language design and implementation, microprocessor systems, data communications, networks, distributed systems, and parallel and novel architectures.

The *Problem Solving* component aims at developing algorithmic thinking, with emphasis on principles of programming and program design. Through this component, students will acquire competence in a number of programming languages using a variety of programming paradigms (imperative, object-oriented, logic-based). Concurrent, parallel, distributed and

heuristic techniques of problem solving are addressed in restricted choices. In this component, students learn how to design, implement and evaluate solutions to significant, albeit relatively small, problems. The wider integration of these techniques in building methodological frameworks to solve real-world problems is studied in courses on systems analysis and design, and software engineering as indicated in the Applications component.

The *Applications* component aims at bringing together the knowledge and skills acquired in the other three components for the development of useful applications to solve "real-world problems". Important technologies such as databases, knowledge bases, graphics and user interface managers are introduced as examples of applications in themselves and as vital tools for the construction of higher level applications. Modern Software Engineering methodologies that address every stage in the planning, design, development, and maintenance of high-quality applications are studied, and subsequently put into practice in the context of a group project undertaken in the framework of the course Software Engineering Professional Practice. Finally, important social and ethical issues concerned with the spread of computers are raised and discussed.

The Undergraduate Program of Studies consists of *Compulsory Courses* covering core material, *Restricted Choices* in Computer Science or related subjects, and *Unrestricted Choices* offered by other Departments. Some courses have other courses as prerequisites. The course dependencies between the compulsory courses are depicted in Table 1.

Since the academic year 2016/2017, the Department of Computer Science is offering the following specialisations (existing students follow the previously offered programme of study):

- Computer Networks
- Fundamentals of Computer Science
- Big Data and Internet Computing
- Real World Computation
- Artificial Intelligence
- Software Engineering
- Digital and Embedded Systems

Each 19pecialization is linked to a number of courses. The aim of the specialisations is to give students the possibility and the motivation to focus on a specific area of Computer Science, via the Elective Courses and an individual diploma project, called undergraduate thesis project. If a student selects at least 3 Elective Courses from the courses indicated for each specialization and undertakes an undergraduate thesis project from the same specialization under the supervision of a faculty member then the respective specialization will be indicated in her transcripts. Applications for the recognition of the specialization need to be submitted by the student, in the dedicated form, to the Undergraduate Program Committee after the period of changes in enrollments in the last semester of studies. The Department is not obliged to offer all restricted choice courses, in order to cover all specializations, nor will it change the restrictions regarding the class size and the process for selecting the undergraduate thesis project.

Almost all courses offered by the Department in the first four semesters are compulsory and common to all students. Students are asked to choose a direction at the beginning of the fourth semester of their studies. In the fifth and sixth semesters, students are required to take the compulsory courses of their chosen direction, as well as elective courses from other departments. The last two semesters include restricted elective courses (three courses from the chosen direction and any two courses in Computer Science 300 or 400 level) within the direction and elective courses; also, each student must undertake an individual Diploma Project, under the supervision of a member of the Department's academic staff. The topic of the Undergraduate Thesis Project must relate to the student's chosen direction of study..

Curriculum

Each course code has the form XYZ, where X represents the type of the course and Y the area it belongs to. Courses offered for students of other Departments have, by default, type 0. Compulsory courses are of 1st, 2nd or 3rd type, restricted elective courses are of 4th type, while the Diploma Project is of 4th type. Course areas have code 1 (Theory), 2 (Computer Systems), 3 (Troubleshooting) and 4 (Applications). General content courses have area code 0.

Below we cite the indicative programs offered by the Department since the academic year 2019/2020, as well as the courses linked with each 20pecialization.

.

Indicative Program

Semester	Curriculum	ECTS
First Semester	CS111 Discrete Structures in Computer Science and Computation	7,5
	CS131 Programming Principles	7,5
	MAS012 Calculus I	5
	LAN100 General Advanced English	5
	Elective Course	5
Second Semester	CS121 Digital Systems	7,5
	CS133 Object Oriented Programming	7,5
	MAS013 Calculus II	5
	LAN111 English for Computer Science	5
	MAS029 Linear Algebra	5
Third Semester	CS221 Computer Organization and Assembly Programming	7,5
	CS231 Data Structures and Algorithms	7,5
	CS232 Programming Techniques and Tools	7,5
	MAS055 Probability and Statistics	7
	CS202 Explorations into Computer Science	3
Fourth Semester	CS211 Theory of Computation	7,5
	CS222 Operating Systems	7,5
	CS236 Algorithms and Complexity	7,5
	Elective Course	5
Fifth Semester	CS324 Communications and Networks	7,5
	CS342 Database Systems	7,5
	CS343 Software Engineering	7,5
	Elective Course	7,5
Sixth Semester	CS325 Parallel Processing	7,5
	CS326 Systems Security	7,5
	CS341 Artificial Intelligence	7,5
	Restricted Elective Course	7,5
Seventh Semester	CS400 Diploma Project I	5
	Restricted Elective Course	7,5
	Restricted Elective Course	7,5
	BPA 369 Principles of Entrepreneurship and Innovation	5
	Elective Course	
Eighth Semester	CS401 Diploma Project II	10
	Restricted Elective Course	7,5
	Restricted Elective Course	7,5
	Elective Course	5

Elective Courses from
Physics, Chemistry, and Biological science departments

BIO102	Integrative Biology of Organisms
PHY131	General Physics I: Mechanics, Waves and Thermodynamics
PHY132	General Physics II: Electricity, Electromagnetic and Optics
CHE121	Introduction to Chemistry

Elective Courses for Specialisation: Computer Networks

CS344	Internet Technologies
CS421	Systems Programming
CS422	Advanced Networks
CS423	Network and Information Security
CS427	Mobile Computer Networks
CS428	Internet of Things: Programming and Applications
CS432	Distributed Algorithms
CS450	Network Virtualisation and Management

Elective Courses for Specialisation: Fundamentals of Computer Science

CS412	Logic in Computer Science
CS414	Basic Principles of Programming Languages
CS431	Synthesis of Parallel Algorithms
CS432	Distributed Algorithms
CS433	Constraint Programming and Satisfaction

Elective Courses for Specialisation: Big Data and Internet Computing*

CS344	Internet Technologies
CS421	Systems Programming
CS446	Advanced Topics in Databases
CS448	Data Mining on the Web
CS450	Network Virtualisation and Management
CS481	Software Engineering for Software as a Service
MAS458	Statistical Data Analysis

Elective Courses for Specialisation: Real World Computation

CS426	Computer Graphics
CS435	Human Computer Interaction
CS444	Computational Intelligent Systems
CS445	Digital Image Processing
CS447	Computer Vision

Elective Courses for Specialisation: Artificial Intelligence

CS412	Logic in Computer Science
CS433	Constraint Programming and Satisfaction
CS434	Logic Programming and Artificial Intelligence
CS442	Machine Learning
CS444	Computational Intelligent Systems
CS445	Digital Image Processing
CS447	Computer Vision
CS448	Data Mining on the Web

Elective Courses for Specialisation: Software Engineering

CS344	Internet Technologies
CS421	Systems Programming
CS435	Human Computer Interaction
CS441	Advanced Software Engineering
CS443	Software Reuse
CS449	Professional Practice in Software Engineering
CS451	Software Analysis
CS481	Software Engineering for Software as a Service

Elective Courses for Specialisation: Digital and Embedded Systems

CS420	Computer Architecture
CS421	Systems Programming
CS428	Internet of Things: Programming and Applications
CS429	Theory and Practice of Compilers
CS445	Digital Image Processing
CS481	Software Engineering for Software as a Service

* Students of specialisation Big Data and Internet Computing can select the course MAS 458 Statistical Data Analysis from the Department of Mathmetics asnd Statistics. This mean that the total number of ECTS will be 0.5 less, since the course has 7 ECTS. The students are encouranged to attend a selective course to cover the 0.5 ECTin their program.

Regarding the distribution of elective courses in the 4th year of their studies in both semesters, the student has the possibility to choose the following alternative program of study, is he/she wishes to:

7th Semester: Diploma Project I
3 Restricted Elective Courses
BPA 369 Principles of Entrepreneurship and Innovation
8th Semester: Diploma Project II
1 Restricted Elective Course
2 Elective Courses

Restricted Elective Courses

In consultation with her/his academic advisor, each student selects Restricted Elective Courses according to her/his interests and professional goals. A student may choose to expand her/his knowledge in one specific area, or may select a combination of courses in more than one area. Thus, the Restricted Elective Courses may be selected so that they satisfy, to a certain significant degree, the student's goals and aptitudes.

Graduate courses offered by the Department of Computer Science may also be selected by the student as Restricted Elective Courses after appropriate approval from the Committee of the Undergraduate Program.

Elective Courses

Each student, in consultation with her/his Academic Advisor, selects elective courses. The selection is made according to the interests and the goals of the student. The Unrestricted Choices must be courses offered by other Departments. According to the Rules for Undergraduate Studies of the University, the elective courses must include courses *from at least three* different Schools of the University. Only one foreign language course of first level is recognized as elective course. If however, a student has succeeded also in the second level course of the same foreign language, both courses are eligible as elective courses.

Foreign Language Courses

Each student must successfully attend two courses in a foreign language. The Department has allocated 10 ECTS credits for these courses and identifies English as a foreign language.

Diploma Project

During the last two semesters of her/his studies, each student undertakes an individual Diploma Project in accordance with regulations approved by the Departmental Board (Meeting of 19/7/95). These regulations have been revised by the Departmental Board (Meeting of 06/12/10) and can be found in Appendix A. The Diploma Projects are collected within the open access Digital Library (http://godigital.cs.ucy.ac.cy).

Minor Program in Computer Science

The Minor Program in Computer Science is open to all students of the University outside the Department of Computer Science. It consists of eight courses with a total workload of

at least 60 credits. Given that schooling starts in Spring, the program can be completed in four consecutive semesters for monitoring two courses per semester. The organization of courses is as follows:

1st Semester (considered as a Fall Semester)
CS131 Programming Principles
CS111 Discrete Structures in Computer Science and Computation

2nd Semester
CS133 Object-Oriented Programming
CS121 Digital Systems

3rd Semester
CS231 Data Structures and Algorithms
One course from core courses or from the restrictive elective courses of the Computer Science curriculum.

4th Semester
Two courses from core courses or from the restrictive elective courses of the Computer Science curriculum.

The number of admitted students in the Minor Program in Computer Science is 10. Necessary criterion for admission is that the weighted average for all previous semesters must be at least 6.5.

Minor Programme in Biomedical Engineering

The Minor Programme in Biomechanics requires the successful completion of ten courses with at least 60 ECTS as a total. It has to be completed between the 8th and 12th semesters of study. The students can choose among the following courses:

Winter Semester (course names appear in Greek)

BIO 102	Αρχές της Βιολογίας I
BIO230	Εισαγωγή στην Υπολογιστική Βιολογία
BIO 442	Πρακτική Άσκηση στη Βιολογία
BIO 495, 496, 497, 498, 499	Σύγχρονα Θέματα Βιολογίας
BIO491/492	Διπλωματική Εργασία I και II
CS 428	Διαδίκτυο των Πραγμάτων: Προγραμματισμός και Εφαρμογές
CS 434	Λογικός Προγραμματισμός και Τεχνητή Νοημοσύνη
CS 435	Αλληλεπίδραση Ανθρώπου-Υπολογιστή
CS 442	Μηχανική Μάθηση
CS 445	Ψηφιακή Επεξεργασία Εικόνας
ECE 421	Ευφυή Συστήματα (ή HMY634)
ECE 429	Εισαγωγή στην Επεξεργασία Ψηφιακών Σημάτων (ή ECE 623)
ECE 434	Εισαγωγή στην Φωτονική (ή ECE 645)
ECE 473	Αισθητήρες & Συστήματα Οργάνων (ή ECE 665)
ECE 476	Βιοϊατρική Απεικόνιση (ή 6xx)
MME 420	Ρομποτική
MME 435	Εισαγωγή στην Βιολογική και Βιοϊατρική Μηχανική
BM XXX	Supervised studies on Biomedical Engineering
BM XXX	Special subjects on on Biomedical Engineering

Spring Semester (course names appear in Greek)

BIO003	Εισαγωγή στη Βιοπληροφορική
BIO 331	Υπολογιστική και Συστημική Βιολογία
BIO 442	Πρακτική Άσκηση στη Βιολογία
BIO 495, 496, 497, 498, 499	Σύγχρονα Θέματα Βιολογίας

CS 341	Τεχνητή Νοημοσύνη
CS 426	Γραφικά Υπολογιστών
CS 444	Συστήματα Υπολογιστικής Νοημοσύνης
CS 447	Υπολογιστική Όραση

ECE 370	Εισαγωγή στη Βιοϊατρική Μηχανική
ECE 425	Ρομποτική
ECE 471	Νευροφυσιολογία και Αισθήσεις (ή ECE 671)
ECE 478	Ψηφιακή Επεξεργασία Εικόνας (ή ECE 626)

MME 436	Μηχανική κυττάρων και ιστών
MME 451	Ανάλυση Στατικής και Δυναμικής Γραμμικής Ελαστικότητας με Πεπερασμένα Στοιχεία

BM XXX	Supervised study on on Biomedical Engineering
BM XXX	Special subjects on on Biomedical Engineering

The students can also enroll to up to 2 from the following master courses:

BIO 630	Νουκλεϊνικά Οξέα
BIO 650	Ειδικά Θέματα Βιοπληροφορικής
BIO 670	Οπτική Απεικόνιση στη Βιολογία

CS 667 / MAI 647	Computational Neuroscience
CS 668	Υπολογιστική Όραση
CS 679	Ηλεκτρονική Υγεία
CS 680	Γνωστικός Προγραμματισμός

ECE 623	Εισαγωγή στην Επεξεργασία Ψηφιακών Σημάτων
ECE 626	Ψηφιακή Επεξεργασία Εικόνας
ECE 627	Μηχανική Όραση
ECE 634	Ευφυή Συστήματα
ECE 645	Εισαγωγή στην Φωτονική
ECE 665	Αισθητήρες & Συστήματα Οργάνων
ECE 671	Νευροφυσιολογία και Αισθήσεις
ECE 6xx	Βιοϊατρική Απεικόνιση

MME 555	Polymers in Medical Applications
MME 531	Μηχανική Συνεχών Μέσων
MME 532	Βιοϋλικά στη Μηχανική Ιστών και την Αναγεννητική Ιατρική

Table 1: Dependencies among courses

Code	Course	Prerequisites / Conditions
CS111	Discrete Structures in Computer Science and Computation	
CS121	Digital Systems	
CS131	Programming Principles	
CS133	Object-Oriented Programming	CS131- Programming Principles
CS202	Explorations into Computer Science	
CS211	Theory of Computation	CS111- Discrete Structures in Computer Science and Computation MAS012- Calculus I
CS221	Computer Organization and Assembly Language Programming	CS121- Digital Systems
CS222	Operating Systems	CS221- Computer Organization and Assembly Language Programming CS232- Programming Techniques and Tools
CS231	Data Structures and Algorithms	CS111- Discrete Structures in Computer Science and Computation CS133- Object-Oriented Programming
CS232	Programming Techniques and Tools	CS131- Programming Principles
CS236	Algorithms and Complexity	CS231- Data Structures and Algorithms
CS324	Communications and Networks	CS131- Programming Principles
CS325	Parallel Processing	CS221- Computer Organization and Assembly Language Programming
CS326	Systems Security	CS232- Programming Techniques and Tools
CS341	Artificial Intelligence	CS231- Data Structures and Algorithms
CS342	Databases	CS231- Data Structures and Algorithms
CS343	Software Engineering	CS133- Object-Oriented Programming
CS344	Internet Technologies	CS133- Object-Oriented Programming
CS421	System Programming	CS222-Operating Systems CS231- Data Structures and Algorithms
CS400 – CS401	Diploma Project	Approval by Academic Advisor
CS412	Logic in Computer Science	CS111- Discrete Structures in Computer Science and Computation
CS414	Basic Principles of Programming Languages	CS211- Theory of Computation CS231- Data Structures and Algorithms
CS420	Computer Architecture	CS222- Operating Systems
CS421	Systems Programming	CS222- Operating Systems
CS422	Advanced Networks	CS324- Communications and Networks
CS423	Network and Information Security	CS324- Communications and Networks
CS424	Digital Signal Processing	CS111- Discrete Structures in Computer Science and Computation MAS016- Linear Algebra for Computer Science MAS012- Calculus I
CS426	Computer Graphics	CS232- Programming Techniques and Tools
CS428	Internet of Things: Programming and Applications	CS221- Computer Organization and Assembly Language Programming CS222- Operating Systems

CS429	Theory and Practice of Compilers	CS211- Theory of Computation CS231- Data Structures and Algorithms
CS431	Synthesis of Parallel Algorithms	CS231- Data Structures and Algorithms
CS432	Distributed Algorithms	CS211- Theory of Computation CS231- Data Structures and Algorithms
CS433	Constraint Programming and Satisfaction	CS111- Discrete Structures in Computer Science and Computation CS231- Data Structures and Algorithms
CS434	Logic Programming and Artificial Intelligence	CS111- Discrete Structures in Computer Science and Computation
CS435	Human - Computer Interaction	
CS441	Advanced Software Engineering	CS343- Software Engineering
CS442	Machine Learning	CS231- Data Structures and Algorithms
CS443	Software Reuse	CS343- Software Engineering
CS444	Computational Intelligence Systems	
CS445	Digital Image Processing	CS231- Data Structures and Algorithms MAS016- Linear Algebra for Computer Science
CS446	Advanced Topics in Databases	CS342- Databases
CS447	Computer Vision	CS231- Data Structures and Algorithms MAS016 Linear Algebra for Computer Science
CS448	Data Mining on the Web	CS231- Data Structures and Algorithms CS342- Databases
CS449	Software Engineering Professional Practice	CS343- Software Engineering
CS450	Network Virtualisation and Management	CS324- Communications and Networks
CS451	Software Analysis	CS211 - Theory of Computation CS232 - Programming Techniques and Tools
CS481	Software Engineering for Software as a Service	CS231- Data Structures and Algorithms CS343- Software Engineering
CS482	Software Validation, Verification and Quality	CS343- Software Engineering
CS483	Software Analysis, Modelling and Design	CS343- Software Engineering
CS499	Special Issues in Computer Science	*Varies*
MAS013	Calculus II	MAS012- Calculus I

Short Description of Courses

Each course description includes the name of the instructor that offers the course in the current academic year 2022-2023. For the restricted courses that are not offered in this academic year, the name of the instructor who suggested the course is included instead. The language of instruction for most courses is **Greek**, whereas a small number of courses are offered in English. Courses taught in English are indicated.

CS111 Discrete Structures in Computer Science and Computation
Type: Compulsory for all direcitons
Level: Undergraduate
Semester: Fall
Credit: 7,5 ECTS units
Instructor: A. Pieris
Objectives: Introduction to basic mathematical concepts that are applicable to Computer Science. Development of mathematical way of thinking about problems. Provision of some necessary mathematical tools.
Content: Foundations: sets and functions. Logic: Propositional Logic: basics of Predicate Logic. Mathematical Reasoning: methods of proof, induction. Counting: basics of counting, pigeonhole principle, permutations and combinations. Relations: properties and applications, equivalence relations, partial orders. Graphs: basic concepts.
Prerequisites: ----------
Bibliography:

1. K. Rosen, *Discrete Mathematics and its Applications*, 5th Edition, McGraw-Hill, 2003.

Teaching methods: Lectures (3 hours weekly) and Recitation (1 hour weekly).
Assessment: Final exam, midterm exam and homework.

CS121 Digital Systems
Type: Compulsory
Level: Undergraduate
Semester: Fall
Credit: 7,5 ECTS units
Instructor: C. Pattichis
Objectives: Introduce various representations of information in a digital computing system. Presentation of the building blocks of digital computing systems. Teach the basic methods for the design and analysis of digital systems. Practical implementation of a simple digital computing system.
Content: Principles of design and construction of digital electronic systems and computers. Representation of data with binary sequences. Data storage and processing by electronic digital circuits. Consolidation of theoretical knowledge through practical exercises in the design and construction of digital circuits in the laboratory for Digital Systems Design and Microprocessors.
Prerequisites: ----------
Bibliography:

1. M. Mano, D.Kime, *Logic and Computer Design Fundamentals and Xilinx 4.2 Package*, 3rd Edition, Prentice Hall, 2003.

Teaching Methods: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam, quizzes and homework (final project and exercises).

CS131 Programming Principles
Type: Compulsory
Level: Undergraduate
Semester: Fall and Spring
Credit: 7.5 ECTS units
Instructor: E.Keravnou-Papailiou (fall /spring)
Objectives: Introduction of methods for problem-solving through programming.

Development of procedural and object-oriented problem solving skills and algorithmic thinking. Provision of deep understanding of basic programming principles and algorithmic techniques, design, implementation, testing and debugging of modular programs. Understanding the important concepts of program abstraction and data abstraction. Mastering of a high-level programming language (Java).
Content: Presentation of the software development process and introduction to the basic principles of programming and program design using the Java language. Global overview of the Java language with emphasis on built-in and abstract data types, control structures, functions, modular programming and reusability.
Prerequisites: ----------
Bibliography:

1. R. Sedgewick, K. Wayne, *Introduction to Programming in Java: An Interdisciplinary Approach*, Addison Wesley, 2008.

Teaching methods: Lectures (3 hours weekly), Laboratory sessions (3 hours weekly) and Recitation (1 hour weekly).
Assessment: Final exam, midterm exam, homework (programming assignments) and quizzes.

CS133 Object Oriented Programming
Type: Compulsory
Level: Undergraduate
Semester: Fall and Spring
Credit: 7.5 ECTS units
Instructor: M. Dikaiakos (Spring) / ……. (Fall)
Objectives: This course provides students with an extended introduction to Object Oriented Programming principles with Java. Students are expected to develop advanced programming and problem-solving skills, applying Object Oriented principles and advanced programming techniques in Java.
Content: Use of object oriented methodology for problem solving. Abstraction and information hiding. Packages. Strings, CRC cards and class diagrams. Inheritance. Polymorphism, Exceptions. Inner classes. Input-output. Type Information. Generics. Collections. Annotations and lambda expressions.
Prerequisites: CS131
Bibliography:

1. W. Savitch, K. Mock. *Absolute JAVA*. 6th Edition, Pearson, 2016.
2. C. S. Horstmann. *Object-Oriented Design & Patterns*. 2nd Edition, Wiley, 2005.
3. R-G. Urma, M. Fusco, A. Mycroft. *Modern Java in Action*. Manning, 2021.
4. B. Eckel, *Thinking in Java*, 4th Edition, Prentice Hall, 2006.
5. *Teaching methods*: Lectures (3 hours weekly), Laboratory sessions (3 hours weekly) and Recitation (1 hour weekly).

Assessment: Final exam, midterm exam and homework.

CS202 Explorations into Computer Science
Type: Compulsory
Level: Undergraduate
Semester: Spring
Credit: 3 ECTS units
Instructor: M. Belk
Objectives: Introduction to topics that compose a global picture of Computer Science. Creation of enthusiasm and interest in Computer Science. Update about current developments in Computer Science. Familiarization with practical applications of Computer Science.
Content: Weekly lecturers/seminars that cover a broad spectrum of Computer Science and its basic areas, starting from its birth and reaching its modern evolutions. Revolutionary ideas for the foundation and development of Computer Science.
Prerequisites: ----------
Bibliography: ----------
Teaching methods: Lectures / Seminars (1.5 hours weekly).

Assessment: Group project and presentation, class participation and attendance.

CS211 Theory of Computation
Type: Compulsory
Level: Undergraduate
Semester: Spring
Credit: 7,5 ECTS units
Instructor: A. Philippou
Objectives: Introduction to foundational concepts of the Theory of Computation. Development and cultivation of formal and syllogistic reasoning. Familiarization with fundamental techniques for proofs and mathematical reasoning. Realization of the limitations on the capabilities of computers.
Content: Formal methods of computation based on machines, grammars and languages: finite automata vs. regular languages; pushdown automata vs. context-free grammars; Turing machines vs. unrestricted grammars. Models of computation equivalent to Turing machines and Church's Thesis. Computability and Uncomputability. Introduction to Theory of Computational Complexity with emphasis on the Theory of NP-completeness.
Prerequisites: CS111, MAS012
Bibliography:

1. M. Sipser, *Εισαγωγή στη Θεωρία Υπολογισμού*, Πανεπιστημιακές Εκδόσεις Κρήτης, 2007
2. M. Mavronicolas, *Θεωρία Υπολογισμού*, προσχέδιο βιβλίου, Αύγουστος 2005.
3. H. R. Lewis, C. H. Papadimitriou, *Elements of the Theory of Computation*, 2nd Edition, Prentice Hall, 1998.

Teaching methods: Lectures (3 hours weekly) and Recitation (1 hour weekly).
Assessment: Final exam, midterm exam and homework.

CS221 Computer Organization and Assembly Language Programming
Type: Compulsory
Level: Undergraduate
Semester: Fall
Credit: 7,5 ECTS units
Instructor: Y. Sazeides / H. Volos
Objectives: Introduction to the basic concepts and methods of computer organization. Present trends of various computer technology parameters. Introduction of the concept of instruction set architecture. Teach different instruction sets and assembly programming. Design of a pipelined datapath. Learn basic principles of memory hierarchy. Implement a simple computing system. Practical experience in the Digital Logic and Microprocessors Laboratory.
Content: Introduction to computer organization and architecture. Types of instructions, coding of instructions, Arithmetic and Logic Unit. Basic principles of the organization of the main functional units of a computer system at machine level: Central Processing Unit (CPU), memory, and Input/Output. Interfacing CPU and peripheral units. Programming in assembly language for MIPS R2000/R3000 and Intel Pentium.
Prerequisites: CS121, CS131
Bibliography:

1. D. A. Patterson, J. L. Hennessy, *Computer Organization and Design - The Hardware/Software Interface*, Third Edition, Morgan Kaufmann, 2005.

Teaching Methods: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final and midterm exams, quizzes and homework (final project and exercises).

CS222 Operating Systems
Type: Compulsory
Level: Undergraduate
Semester: Spring
Credit: 7,5 ECTS units
Instructor: G. Papadopoulos

Objectives: Introduction to the basic principles of design and operation of modern operating systems. Familiarization with the various operation levels and mechanisms, case studies involving typical operating systems like UNIX and Windows as well as the dual role of the operating system, as manager of the various parts of the computer hardware and supplier of offered services to the user.
Content: Introduction, history and evolution of operating systems. General structure, operations and characteristics of an operating system. Concurrency. Process management. Scheduling and dispatch. Real and virtual memory management. I/O management and disk scheduling. File management.
Prerequisites: CS221, CS232
Bibliography:

1. W. Stallings, *Operating Systems: Internals and Design Principles*, 9th Edition, Prentice Hall, 2015.

Teaching methods: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework (theoretical and programming assignments).

CS231 Data Structures and Algorithms
Type: Compulsory
Level: Undergraduate
Semester: Fall and Spring
Credit: 7,5 ECTS units
Instructor: M. Mavronicolas (Fall) / ……. (Spring)
Objectives: Familiarization with data structures and the algorithms manipulating them. Appreciation of the importance of careful organization of information for efficient searching and modification. Acquaintance with techniques for the analysis of algorithm efficiency. Developments of capabilities for designing algorithms to minimize their execution time and space requirements.
Content: Study of data structures for the organization and efficient processing of data. Linear and non-linear data structures. Hashing techniques. Sorting algorithms. Graph algorithms. Introduction to algorithm design techniques. Analysis of the average and worst-case complexity of algorithms.
Prerequisites: CS111, CS132
Bibliography:
1. T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, *Introduction to Algorithms*, 3rd Edition, The MIT Press, 2009.2. M. T. Goodrich, R. Tamassia, M. H. Goldwasser, *Data Structures and Algorithms in JAVA*, 6th Edition, John Wiley & Sons, 2014.
Teaching methods: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory sessions (1.5 hours weekly).
Assessent: Final exam, midterm exam and homework (theoretical and programming assignments).

CS232 Programming Techniques and Tools
Type: Compulsory
Level: Undergraduate
Semester: Fall
Credit: 7,5 ECTS units
Instructor: A. Aristidou
Objectives: The course teaches intermediate and advanced programming concepts, techniques and tools through a language that compiles to machine code. The course familiarizes the students with advanced programming constructs utilized for handling memory and files. Advanced topics in compilation, debugging, documentation and optimization of software. Methodological aspects in developing large-scale system software that addresses complex problems. Basic commands for programmers in the UNIX operating system.
Content: i) Introduction to C for Programmers: types x86/x64, loops, selections, expressions,

arrays, functions, IO, basic program organization, ii) Advanced C programming constructs: program anatomy and processes, memory and addresses (pointers, pointers and arrays, strings and examples), structures, unions and enumerations. Linear and non-linear programming data structures (dynamic memory allocation, lists, queues, doubly-linked lists, trees, applications and examples). iii) Advanced Compilation Topics and Tools: preprocessor directives, compiling multiple files with makefiles, static (.a) and dynamic (.so) linking of object files (.o), error handling (assert.h), static and dynamic code analysis (valgrind and gprof). iv) low-level programming (binary operators and examples, binary files and hexdump). v) Basic commands for programmers in the UNIX operating system: file system, redirection and pipes, permissions and basic filters.
Prerequisites: CS 131
Bibliography:

1. K.N. King, *C Programming: A Modern Approach*, 2nd Edition, ISBN-10: 0393979504, ISBN-13: 978-0393979503, W. W. Norton & Company, 2008.
2. St. G. Kochan *Programming in C*, 4th Edition, ISBN-10: 0321776410, ISBN-13: 9780321776419, Addison-Wesley Professional, 2015.
3. S. Das, *Your UNIX/Linux: The Ultimate Guide*, 3rd Edition, McGraw Hill, ISBN-13 9780073376202, 2013.
4. Ν. Χατζηγιαννάκης, *Η Γλώσσα C σε Βάθος*, Τρίτη Έκδοση, 978-960-461-208-6, Κλειδάριθμος, 2009.

Teaching methods: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam, programming assignments.

CS236 Algorithms and Complexity
Type: Compulsory
Level: Undergraduate
Semester: Spring
Credit: 7,5 ECTS units
Instructor: M. Mavronicolas (Fall) / Chr. Georgiou (Spring)
Objectives: Familiarization with fundamental techniques of designing and analyzing algorithms. Familiarization with significant algorithms in various fields that have been suggested in the literature. Familiarization with techniques for implementing and empirically evaluating algorithms.
Content: Topics in the design and analysis of efficient algorithms and their complexity. Significant algorithms in Graph Theory, Algebra, Geometry, Number Theory and Combinatorics. General algorithmic techniques (e.g., divide-and-conquer, backtracking, dynamic programming). Randomized, Parameterized and Approximation algorithms. Fast Fourier Transform. Inherent lower bounds on problem complexity.
Prerequisites: CS231
Bibliography:

1. J. Kleinberg, E. Tardos, *Σχεδίαση Αλγορίθμων*, Εκδόσεις Κλειδάριθμος, 2008. (English Version: J. Kleinberg, E. Tardos, *Algorithm Design*, Addison Wesley, 2006).
2. S. Dasgupta, C. Papadimitriou, U. Vazirani, *Αλγόριθμοι*, Εκδόσεις Κλειδάριθμος, 2009.

Teaching methods: Lectures (3 hours weekly) and Recitation (2.5 hours weekly).
Assessment: Final exam, midterm exam and homework (theoretical and programming assignments).

CS324 Communications and Networks
Type: Compulsory
Level: Undergraduate
Semester: Fall
Credit: 7,5 ECTS units
Instructor: A. Pitsillides / V. Vassiliou
Objectives: Familiarization with fundamental topics in communication networks, with a focus on the Internet.

Content: Introductory course in Communication Networks. The goal is the understanding and use of concepts related to fundamental issues in Communication Networks, using the Internet as an example. Deals with Networking layers, such as the application, transport, network, and data link layers. Open systems and internetworking. Networking technologies including wired and wireless Local Area Networks and network topologies. Algorithms, including routing and congestion control. Introduction to quality of service (QoS) and multimedia applications. Introduction to emerging networking topics, as e.g. the Internet of Things, Network softwarisation and 5G/5G. Laboratory session includes practical exercises with wireshark and simulations using OPNET.
Prerequisites: CS131
Bibliography:

1. J. F. Kurose, K. W. Ross, *Computer Networking: A Top-Down Approach Featuring the Internet*, 7th Edition, Addison-Wesley, 2016.

Teaching methods: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework (including laboratory exercises).

CS325 Parallel Processing
Type: Compulsory
Level: Undergraduate
Semester: Spring
Credit: 7,5 ECTS units
Instructor: Y. Sazeides / H. Volos
Objectives: Introduction to the fundamental concepts of Parallel Processing with emphasis on parallel machines and parallel programming.
Content: The entire spectrum of parallel machines as appearing in Flynn's classification: SISD, SIMD, MISD, MIMD. The main approaches for design and operation of multiprocessor systems. Conventional and non-conventional machines (Data-flow and reduction). Parallel programming approaches: (1) Automatic-parallelizing compilers, (2) Extending serial languages with parallelizing constructs, (3) parallel languages for Functional Programming. Special emphasis on parallel architectures and parallel programming.
Prerequisites: CS221
Bibliography:

1. D. E. Culler, J. P. Singh, *Parallel Computer Architecture – A Hardware/ Software Approach*, Morgan Keufmann, 1999.

Teaching methods: Lectures (3 hours weekly) and Recitation (1 hour weekly).
Assessment: Final exam, midterm exam, homework and final project.

CS326 Systems Security
Type: Compulsory
Level: Undergraduate
Semester: Spring
Credit: 7,5 ECTS units
Instructor: E. Athanasopoulos
Objectives: Introduction to systems security which covers a wide range of concepts. Primarily, the course helps students to become familiar with different research areas of modern systems security by covering topics such as applied cryptography, software vulnerabilities and exploitation, defenses, mobile security, web security, network security, privacy, and anonymity. Additionally, students will be able to perform engineering tasks in cryptography, software exploitation, network security, and program analysis.
Content: Introduction to applied cryptography (early ciphers, modern symmetric and asymmetric ciphers, cryptographic hash functions and MACs, applications). Software vulnerabilities (buffer overflows, integer overflows, use-after-free). Control-flow attacks (code injection, code reuse). Defenses (non-executable pages, stack canaries, code randomization, CFI). Isolation techniques (SFI). Static and dynamic analysis of software. Mobile security (Android, iOS). Web security (cross-site scripting, CSRF, clickjacking,

phishing). Network security (TLS, botnets, DDoS). Authentication and passwords. Privacy and anonymity (TOR).
Prerequisites: CS232
Bibliography:

1. A. J. Menezes, P. C. van Oorschot, S. A. Vanstone. *Handbook of Applied Cryptography*, CRC Press.
2. R. Anderson. *Security Engineering: A Guide to Building Dependable Distributed Systems*, 2nd Edition, Wiley (http://www.cl.cam.ac.uk/~rja14/book.html)
3. C. Paar, J. Pelzl, *Understanding Cryptography: A Textbook for Students and Practitioners*, Springer.
4. Published papers.

Teaching methods: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework (including laboratory assignments).

CS341 Artificial Intelligence
Type: Compulsory
Level: Undergraduate
Semester: Spring
Credit: 7,5 ECTS units
Instructor: Y. Dimopoulos
Objectives: Introduction to the fundamental principles and techniques that underline software systems that exhibit "intelligent" behavior. Upon completion of this course, students will have acquired a good understanding of modern Artificial Intelligence, the problems that it addresses and the fundamental solution methods that it uses.
Content: Introduction to Artificial Intelligence. Intelligent Agents. Search. Constraint Satisfaction Knowledge Representation and Reasoning. Action Planning. Machine Learning. Uncertainty.
Prerequisites: CS231
Bibliography:

1. S. Russel, P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd Edition, Prentice Hall, 2014.
2. E. Keravnou, *Τεχνητή Νοημοσύνη και Έμπειρα Συστήματα*, Ελληνικό Ανοικτό Πανεπιστήμιο, 2000.

Teaching methods: Lectures (3 hours weekly) and Recitation (1 hour weekly).
Assessment: Final exam, midterm exam and homework (theoretical and programming assignments).

CS342 Databases
Type: Compulsory
Level: Undergraduate
Semester: Fall
Credit: 7,5 ECTS units
Instructor: D. Zeinalipour
Objectives: Introduction to the basic principles needed for the design and the use of ta database. Provision of practical exercises in the application of these concepts with the use of an industrial system for database.
Content: Introduction to Databases. Organization and proper management of large quantities of data for use in applications. Database models such as the entity-relation model, the relational model, the network model and the hierarchical model.
Prerequisites: CS231
Bibliography:

1. R. Elmarsi, S. Navathe, *Fundumentals of Database Systems*, 5th Edition, Addison-Wesley, 2007.
2. R. Elmarsi, S. Navathe, *Θεμελιώδεις Αρχές Συστημάτων Βάσεων Δεδομένων*, 5η Έκδοση, 1ος τόμος, Εκδόσεις Δίαυλος, 2007.
3. R. Ramakrishnan, J. Gehrke, *Database Management Systems*, 3rd Edition, McGraw-Hill, 2003.

1. *Teaching methods*: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory sessions (1.5 hours weekly).
   Assessment: Final exam, midterm exam and homework (theoretical assignments, final project).

CS343 Software Engineering
Type: Compulsory
Level: Undergraduate
Semester: Fall
Credit: 7,5 ECTS units
Instructor: G. Kapitsaki
Objectives: Familiarization with and assimilation of the approaches, methodologies, models and tools used to develop quality software systems. Understanding of software architectures, software modeling and testing. Understanding of software testing process and of software architecture design patterns. Applications of software creation methodologies on the construction of a real software system. Understanding of software testing procedures. Use and application of architecture design patterns.
Content: Methods, tools, and procedures for the development and maintenance of large-scale software systems. Existing life-cycle models (e.g. waterfall model). Introduction to Agile development. Requirements analysis and specification techniques. Software development methodologies. Unified Modelling Language (UML) and supported static and dynamic diagrams. Code transformation. Practical experience with CASE tools for modeling data and procedures (Modelio). Prototyping for Web applications (HTML, CSS). Architectural Design patterns (Model View Controller etc.). Software verification and validation. Unit testing and frameworks (JUnit etc.). CASE tools. Project planning and management.
Prerequisites: CS133
Bibliography:
1. I. Sommerville, *Software Engineering*, 10th Edition, Addison-Wesley, 2016.
2. R. Pressman, *Software Engineering: A Practitioner's Approach*, 8th Edition, Mc Graw Hill, 2015.
3. H. van Vliet, *Software Engineering: Principles and Practice*, 3rd edition, John Wiley & Sons, 2008
4. P. Stevens, R. Pooley, *Using UML Software Engineering with Objects and Components*, 2nd edition, Addison-Wesley, 2006.
   Teaching methods: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory sessions (1.5 hours weekly).
   Assessment: Final exam, midterm exam and semester team project.

CS344 Internet Technologies
Type: Restricted choice
Level: Undergraduate
Semester: Fall or Spring
Credit: 7,5 ECTS units
Instructor: ----------
Objectives: Introduction to Internet and the World-Wide Web Technologies. Emphasis on WWW protocols and programming of Internet Systems and Services.
Content: Internet Fundamentals, Principles of Hypertext, HTML and Mark-up Languages, Cascading Style Sheets, Page Layouts, Web Design & Development, Dynamic content and HTML Forms, Client-Side Programming, Introduction to Web2.0 principles and AJAX,
Prerequisites: CS133, CS324
Bibliography:
1. A. Moller, M. Schwartzbach, *An Introduction to XML and Web Technologies*, Addison Wesley, 2006.

Teaching methods: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam, and homework assignments.

CS412 Logic in Computer Science
Type: Restricted choice
Level: Undergraduate
Semester: Fall
Credit: 7,5 ECTS units
Instructor: A. Philippou
Objectives: The main objective of the course is to prepare students for using logic as a formal tool in Computer Science. Furthermore, it aims to develop and cultivate formal and syllogistic reasoning and provide a thorough introduction to computational logic and its applications in Computer Science.
Content: Propositional Logic Syntax, Semantics, Normal Forms, Decision Procedures, Proof Theory, Compactness and Resolution. Predicate Logic: Syntax and Semantics, Proof Theory Soundness and Completeness and Resolution. Logic Programming. Programming Language Semantics and Verification. Linear and Branching Temporal Logics: Syntax, Semantics, and model-checking algorithms.
Prerequisites: CS111
Bibliography:

1. M. Huth, A. Ryan, *Logic in Computer Science: Modeling and Reasoning about Concurrent Systems*, Cambridge University Press, 2000.
2. M. Ben-Ari, *Mathematical Logic for Computer Science*, Springer-Verlag, 2nd Edition, 2003.
3. U. Schoning, *Logic for Computer Scientists*, Springer-Verlag, 2nd Printing, 2008.

Teaching methods: Lectures (3 hours weekly) and Recitation (1 hour weekly).
*Assessmen*t: Final exam, midterm exam and homework.

CS414 Basic Principles of Programming Languages
Type: Restricted choice
Level: Undergraduate
Semester: ----------
Credit: 7,5 ECTS units
Instructor: A. Philippou
Objectives: The objective of the course is to study the basic principles of programming languages through the representation of basic programming notions into a uniform mathematical framework. At the same time, the course will follow the practical application of the notions under study into popular programming languages. In particular, the course will study the abstract syntax, the syntax, semantics, and the practical application of functional, procedural, object-oriented, and programming structures. Attention will be given to the static (type systems) and dynamic (execution behavior) properties of programming languages, as well as the notions of type soundness and execution safety. The course will introduce principles/notions of distributed programming and study modern execution safety problems in a distributed setting.
Content: History of Programming Languages. λ – calculus. Functional Programming (The Haskell programming language, Haskell type system, Haskell's Type Inference, Pure vs impure programming languages). Procedural and Object-Oriented Programming (Basic object-oriented programming notions, Mathematical representation of the Java programming language). Distributed programming (Actors programming model, Process Calculi as mathematical models for representing distributed programming).
Prerequisites: CS211, CS231
Bibliography:

1. B. C. Pierce, *Types and Programming Languages*, The MIT Press.
2. G. Hutton, *Programming in Haskell*, 2nd Edition, Cambridge University Press, 2016.
3. G. Agha, *Actors: A Model of Concurrent Computation in Distributed Systems*, The MIT Press.
4. At. Igarashi, B. C. Pierce, P. Wadler, *Featherweight Java: A Minimal Core Calculus for Java and GJ*, ACM Trans. Program. Lang. Syst., 23(3), 2001.

Teaching methods: Lectures (3 hours weekly) and Recitation (1 hour weekly).

*Assessmen*t: Final exam, midterm exam and homework.

CS420 Computer Architecture
Type: Restricted choice
Level: Undergraduate
Semester: Fall
Credit: 7,5 ECTS units
Instructor: Y. Sazeides
Objectives: Introduction to current methodology for performance evaluation and comparison of computer systems. Presentation of basic and advanced concepts in the organization of modern microprocessors. Examination of current trends in the computer architecture area.
Content: Introduction to the state-of-the-art uniprocessor, high performance computer architecture. Emphasis on quantitative analysis and cost/performance trade-offs in the design of the basic units of a processor: instruction set, pipelining, memory system and input/output systems. Qualitative analysis of real machines and their performance data.
Prerequisites: CS222
Bibliography:

1. J. Henessy, D. Patterson, *Computer Architecture: A Quantitative Approach.* 3rd Edition. Morgan Kaufmann, 2002.
2. Selected research articles

Teaching Methods: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework (project and exercises).

CS421 Systems Programming
Type: Restricted choice
Level: Undergraduate
Semester: Spring
Credit: 7,5 ECTS units
Instructor: D. Zeinalipour
Objectives: The main objective of this undergraduate course is to allow students develop complex system-level software in the C programming language while gaining an intimate understanding of the UNIX operating system (and all OS that belong to this family, such as Linux, the BSDs, and even Mac OS X). Topics covered will include the user/kernel interface, fundamental concepts of UNIX, user authentication, basic and advanced I/O, filesystems, signals, process relationships, and inter-process communication. Fundamental concepts of software development and maintenance on UNIX systems will also be covered. The taught concepts have an application to the whole family of UNIX OSs (e.g., Linux, MacOS, HP-UX, AIX, Solaris, Android, iOS, Raspbian) but also apply to Windows (e.g., Powershell and windows system calls).
Contents: i) Advanced commands of the UNIX operating system for administrators: filters with regular expressions, system utilities and stream editors (awk, sed). ii) Advanced shell programming with an emphasis on Bash: environment, control structures, debugging. iii) Low-level I/O system calls with C. iv) Process management: environment, control and signals, inter-process communication (IPC) with an emphasis on pipes and named pipes (FIFO), XSI IPC: semaphores, shared memory and message queues, network IPC (TCP Sockets) and the client/server model. v) Multithreaded programming, concurrency and performance aspects. Implementing network protocols from RFC documents. v) System security aspects, managing cluster computers and clouds: virtualization, data centers and Green IT.
Prerequisites: CS222
Bibliography:

1. S. Das, *Your UNIX/Linux: The Ultimate Guide*, 3rd Edition, McGraw Hill, 2013.
2. E. Nemeth, G. Snyder, T. R. Hein, B. Whaley, *UNIX and Linux System Administration Handbook*, 4th Edition, ISBN-10: 0131480057, ISBN-13: 9780131480056, Prentice Hall, 2011.
3. R. E. Bryant, D. R. O'Hallaron, *Computer Systems: A Programmer's Perspective*, 3rd Edition, Pearson, 2016.

4. W. R. Stevens, St. A. Rago, *Advanced Programming in the UNIX Environment*, 3rd Edition, Addison-Wesley Professional, 2013.

1. *Teaching Methods*: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory sessions (1.5 hours weekly).
   Assessment: Final exam, midterm exam, programming assignments.

CS422 Advanced Networks
Type: Restricted choice
Level: Undergraduate
Semester: Spring
Credit: 7,5 ECTS units
Instructor: V. Vassiliou / A. Pitsillides
Language: English
Objectives: Extension of the basic knowledge about Computer Networks regarding architectures, techniques and protocols for the Internet.
Content: Advanced topics in Computer Networks and the Internet, such as: IPv6, Multicast Routing, QoS Routing, TCP Congestion Control, Performance Analysis, Multimedia Networking Applications, Realtime services and protocols, Quality of Service, MPLS, Traffic Engineering, Mobile and Wireless Networks, Issues in Security for Computer Networks. Introduction to advanced research topics (e.g. Internet of Things, wireless sensor networks, VANETs, 5G, Nanonetworks). Introduction to Network Management, Software Defined Networks. Cloud and Fog Computing.
Prerequisites: CS324
Bibliography:

1. J. F. Kurose, K. W. Ross, *Computer Networking – A Top Down Approach Featuring the Internet*, 7th Edition, Addison-Wesley, 2016.
2. Y.-D. Lin, T.-H. Hwang, F. Baker, *Computer Networks, An Open Source Approach* , McGraw Hill, 2011

Teaching methods: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework (including laboratory exercises).

CS423 Network and Information Security
Type: Restricted choice
Level: Undergraduate
Semester: ----------
Credit: 7,5 ECTS units
Instructor: V. Vassiliou
Objectives: Introduction to network and information security principles, understanding of basic areas in Cryptography, Authentication and Confidentiality. Gain of knowledge in methods for the evaluation of Software, Applications and Systems with respect to security. Application of tools for the protection of networks, applications and information.
Content: Introduction to Security Threats and Attacks, Cryptographic Techniques (encryption, cryptanalysis, authentication, confidentiality), identification and authentication (Kerberos, PKI), Internet Application security protocols (PGP, SSL/TLS), Network security (Firewalls, IDS), Defending against threats on endsystems, Checking of networks and applications for vulnerabilities, Other issues in network and information security (privacy, ethics, legal framework).
Prerequisites: CS324
Bibliography:

1. W. Stallings, *Network Security Essentials*, 3rd Edition, Pearson-Prentice Hall, 2007.
2. C.P. Pfleeger, S.L. Pfleeger, *Security in Computing*, 4th Edition, Prentice Hall, 2006.

Teaching methods: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework.

CS424 Digital Signal Processing

Type: Restricted choice
Level: Undergraduate
Semester: ……
Credit: 7,5 ECTS units
Instructor: A. Pitsillides
Objectives: Introduction to Digital Signal Processing (DSP) methods and applications.
Content: Discrete signals and systems, sampling of signals, frequency analysis of discrete systems and signals, z-transform, Fourier-Transform, Discrete Fourier Transform, and Fast Fourier Transform, digital filters, application examples.
Prerequisites: CS111, MAS016, MAS017
Bibliography:

1. S. Theodorides, *Digital Signal Processing*, University of Patras Publishing, 1992.
2. J. H. McClellan, R. W. Schafer, M. A. Yoder, *DSP First*, Prentice Hall, 1998.
3. The Student Edition of Matlab: *User's, Guide*, Prentice Hall, 2005.

Teaching methods: Lectures (3 hours weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework (laboratory exercises, additional exercises, final project).

CS426 Computer Graphics
Type: Restricted choice
Level: Undergraduate
Semester: Fall or Spring
Credit: 7,5 ECTS units
Instructor: A. Aristidou
Objectives: Introduction to the basic principles of digital image synthesis. Explain how a 3-dimensional virtual world is defined starting from the geometry, the materials, the lights and cameras and how the 2-dimensional resulting image is produced by going through the graphics pipeline. Provision of both the theoretical foundations as well as practical skills through the use of industry standards, such as OpenGL or DirectX. Finally, the students will learn to use popular game consoles to develop an application, such as Unity3D
Content: Scene construction, scene hierarchies, camera specification, projections of primitives, clipping, visible surface determination, polygon rasterisation (z-buffer), texture mapping, local and global illumination, shadows, ray tracing, radiosity, real-time acceleration techniques, AR/VR, introduction to character animation.
Prerequisites: CS232
Bibliography:

1. J. F. Hughes, A. van Dam, M. Mcguire, D. F. Sklar, J. D. Foley, S. K. Feiner, K. Akeley, *Computer Graphics: Principles and Practice*, Addison-Wesley Professional; 3rd edition, ISBN-13: 978-0321399526, 2013.
2. M. Slater, A. Steed, Y. Chrysanthou, *Computer Graphics and Virtual Environments: From Realism to Real-Time*, Addison Wesley publishers, ISBN 0-201-62420-6, 2001.
3. T. Theoharis, G. Papaioannou, N. Platis, N. M. Patrikalakis, A K Peters, *Graphics and Visualization: Principles & Algorithm*, ISBN-13: 978-1568812748, 2007.
4. E. Angel, D. Shreiner, *Interactive Computer Graphics with WebGL* Pearson, 7th edition, ISBN-13: 978-1292019345. 2014.

Teaching methods: Lectures (3 hours weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework.

CS427 Mobile Computer Networks
Type: Restricted Choice
Level: Undergraduate
Semester: Fall or Spring
Credit: 7,5 ECTS units
Instructor: A. Pitsillides / V. Vassiliou
Objectives: The objective of this course is to introduce students into wireless

mobile/local/cellular networks with an emphasis on the fundamental concepts and principles of the technology which are important for the design, application, evaluation and development of these systems. The course will also cover new architectures and topologies, existing and proposed standards as well as open research issues.
Content: Wireless environment, Interference and other problems in wireless communications, Architectures and technologies of wireless networks and wireless communication, Wireless MAC protocols, Wireless Local Area Networks (WLAN), Mobile Networks (emphasis on latest architectures, 3G, 4G, 5G), Mobility Management, wireless network technologies (ad-hoc, sensor, vehicular networks, Internet of Things), Open research issues and challenges.
Prerequisites: CS324
Bibliography:

1. D. P. Agrawal, Q.-A. Zeng, *Introduction to Wireless and Mobile Systems*, 3rd Edition, CL-Engineering; 3rd Edition, 2010.
2. W. Stallings, *Wireless Communications and Networks*, Prentice Hall, 2nd Edition, 2002.
3. M. Sauter, *From GSM to LTE: An Introduction to Mobile Networks and Mobile Broadband*, John Wiley & Sons, Ltd., 2011.
4. J. Schiller, *Mobile Communications*, 2nd Edition, Addison-Wesley, 2003.

1. *Teaching methods*: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory sessions (1.5 hours weekly).
   Assessment: Final exam, midterm exam and homework.

CS428 Internet of Things: Programming and Applications
Type: Restricted Choice
Level: Undergraduate
Semester: Fall or Spring
Credit: 7,5 ECTS units
Instructor: C. Pattichis
Language: English
Objectives: ----------
Content: Programming of embedded systems. Introduction to the Internet of Things. Introduction to the mobile processors of ARM for embedded systems. Interruptions and memory consumption. Programming for embedded and mobile systems. Mobile computing. Architecture of mobile processors of ARM. Programming in smartphones and applications. Network connectivity: Bluetooth. Cloud Computing. Connectivity on the cloud.
Prerequisites: CS221, CS222
Bibliography:

1. W. Woff, *Computers as Components: Principles of Embedded Computing System Design*, Morgan Kaufman.

Teaching methods: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework.

CS429 Theory and Practice of Compilers
Type: Restricted choice
Level: Undergraduate
Semester: Fall or Spring
Credit: 7,5 ECTS units
Instructor: E. Athanasopoulos
Objectives: Introduction to advanced techniques of compiler design and implementation.
Content: Fundamental principles of compiler design. Relation of translators to formal languages and automata theory. Lexical, syntactic and semantic analysis, code generation and optimization, etc. Practical exercises using lex and yacc.
Prerequisites: CS211, CS231
Bibliography:

1. V. Aho, R. Sethi, J. D. Ullman, *Compilers – Principles, Techniques, and Tools*, Addison-Wesley, 1986.

Teaching methods: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework (theoretical and diagnostic assignments and semester project).

CS431 Synthesis of Parallel Algorithms
Type: Restricted choice
Level: Undergraduate
Semester: Fall or Spring
Credit: 7,5 ECTS units
Instructor: Chr. Georgiou
Objectives: Introduction to the fundamental techniques of parallel algorithm design and the use of these techniques in designing and analyzing parallel algorithms for basic problems. Introduction to the basic failure types appearing in parallel computing and the appropriate fault-tolerance techniques.
Content: Introduction to parallel computing. Complexity and efficiency measurements of parallel algorithms. Parallel computing models. Basic techniques for the design of parallel algorithms. Efficient parallel algorithms in Combinatorics, Graph Theory, and Matrix Theory. Complexity analysis of algorithms on the Parallel Random Access Machine (PRAM). Comparison between various models of computation. Advanced topics (fault-tolerance, atomicity, synchronization, computational limitations of PRAM).
Prerequisites: CS231
Bibliography:
1. J. Jaja, *An Introduction to Parallel Algorithms*, Addison-Wesley, 1992.
2. R. Miller, L. Boxer, Algorithms *Sequential & Parallel: A Unified Approach*, Prentice Hall, 2000.

Teaching methods: Lectures (3 hours weekly) and Recitation (1 hour weekly).
Assessment: Final exam, midterm exam and homework (theoretical assignments).

CS432 Distributed Algorithms
Type: Restricted choice
Level: Undergraduate
Semester: Fall or Spring
Credit: 7,5 ECTS units
Instructor: M. Mavronicolas
Objectives: Familiarization with the fundamental concepts of the Theory of Distributed Computing. Development of capabilities of designing, proving correct and analyzing distributed algorithms. Cultivation of syllogistic and mathematical approach to the field of distributed algorithms.
Content: Formal models of distributed computing: shared memory versus message passing, determinism versus randomization, concepts of synchronism, asynchrony and real-time. Design and analysis of distributed algorithms and impossibility/improbability results for fundamental problems such as mutual exclusion, consensus, synchronization, leader election, construction of minimum spanning trees. Fault tolerance: Byzantine generals, wait-free algorithms, fault degrees. Formal methods for proving correctness of distributed algorithms. Advanced topics. Special emphasis throughout the course on lower and upper bounds on time and memory.
Prerequisites: CS211, CS231
Bibliography:
1. H. Attiya, J. L. Welch, *Distributed Computing: Fundamentals, Simulations and Advanced Topics*, Second Edition, John Wiley and Sons, 2003.

Teaching methods: Lectures (3 hours weekly) and Recitation (1 hour weekly).
Assessment: Final exam, midterm exam, homework, participation in class and attendance.

CS433 Constraint Programming and Satisfaction
Type: Restricted choice
Level: Undergraduate
Semester: Fall or Spring

Credit: 7,5 ECTS units
Instructor: Y. Dimopoulos
Objectives: A significant number of problems in Computer Science over a wide spectrum are special cases of Constraint Satisfaction problem. This course introduces ways to approach such problems and the associated software. Students will be able to understand the structure and the behavior of Constraint Satisfaction problems and will get exposure to basic algorithms solving them. They will get experience over tools for Constraint Programming, the range of solvable problems and their applications to problem solving.
Content: Definition of constraint satisfaction problems. Constraint representation and complexity. Various forms of consistency. Backtracking, look-ahead and conditions for efficient solving. Algorithms for Propositional Satisfiability. Available commercial products. Study of problems from different application domains, their modeling and solution methods.
Prerequisites: CS111, CS231
Bibliography:

1. F. Rossi, P. van Beek, T. Walsh. *Handbook of Constraint Programming.* Elsevier, 2006.
2. R. Dechter, *Constraint Processing*, Morgan Keufmann, 2003.

Teaching methods: Lectures (3 hours weekly) and Recitation (1 hour weekly).
Assessment: Final exam, midterm exam and homework (programming assignments).

CS434 Logic Programming and Artificial Intelligence
Type: Restricted choice
Level: Undergraduate
Semester: Fall or Spring
Credit: 7,5 ECTS units
Instructor: A. Kakas
Language: English
Objectives: Familiarization with the basic concepts of Logic Programming and practical exercises in implementing them with the PROLOG language. Development of capabilities of applying Logic Programming to problems of Artificial Intelligence.
Content: Basic principles of Logic Programming and implementation using the language Prolog. Relation of Logic Programming to modern considerations regarding Artificial Intelligence. Solving application problems drawn from the fields of Artificial Intelligence and the Semantic Web, making use of Logic Programming and Constraint Logic Programming.
Prerequisites: CS111
Bibliography:

1. L. Sterling, E. Shapino, *The Art of Prolog*, 2nd Edition, The MIT Press, 1994.
2. M. Bramer, *Logic Programming with Prolog*, Springer, 2009.

Teaching methods: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory sessions (1 hour weekly).
Assessment: Final exam, midterm exam and homework.

CS435 Human Computer Interaction
Type: Restricted choice
Level: Undergraduate
Semester: Spring
Credit: 7,5 ECTS units
Instructor: G. Papadopoulos / ……..
Objectives: Appreciation of the importance of designing good user interfaces and the relation of user interface design and the way users interact with computers. Experience with applying a well-known methodology for designing interactive systems, starting from identifying user's needs, to applying usability evaluation methods.
Content: Analysis of the human as a computer system user (knowledge models, graphical animation, cognitive models). Interactive technologies (input-output devices, window environments, systems for collaborative support, virtual reality). Methodologies for the design of interactive systems.
Prerequisites: ----------

Bibliography:

1. Ν. Αβούρης, *Εισαγωγή στην επικοινωνία ανθρώπου-υπολογιστή*, Εκδόσεις ΔΙΑΥΛΟΣ, Αθήνα, 2000.
2. J. Preece, Y. Rogers, H. Sharp, I*nteraction Design: Beyond Human-Computer Interaction*, John Wiley and Sons, 2002.

Teaching methods: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory sessions (1.5 hours weekly).

Assessment: Final exam, midterm exam and homework.

CS441 Advanced Software Engineering

Type: Restricted Choice
Level: Undergraduate
Semester: Spring
Credit: 7,5 ECTS units
Instructor: G. Papadopoulos
Language: English
Objectives: Familiarization and understanding of advanced principles, concepts and practices of software engineering. A number of contemporary areas of software engineering will be covered. This course further serves as a "roadmap" for advanced electives and graduate courses in software engineering.
Content: Software reuse. Distributed software engineering. Service oriented architectures. Real-time systems. Project Management. Human-Computer Interaction for Software Engineering (interaction design, design rules, implementation and evaluation, universal design, user support).
Prerequisites: CS343
Bibliography:

1. I. Sommerville, *Software Engineering*, 10th Edition, Addison-Wesley, 2016.
2. A. Dix, J. Finlay, G. D. Abowd, R. Beale, *Human Computer Interaction*, 3rd Edition, Prentice Hall, 2004.

Teaching methods: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory sessions (1.5 hours weekly).

Assessment: Final exam, midterm exam, homework (theoretical problems – study / analysis of a software system) and laboratory exercise.

CS442 Machine Learning

Type: Restricted Choice
Level: Undergraduate
Semester: Fall
Credit: 7,5 ECTS units
Instructor: Chr. Christodoulou
Objectives: Familiarization with theoretical and practical issues involved in Machine Learning. Study of machine learning methods as they have been developed in recent years. Implementation and assessment of Machine Learning systems.
Content: Introduction to Pattern Recognition, Multilayered Neural Networks and backpropagation learning algorithm, Deep Learning and Convolutional Neural Networks, Recurrent Neural Networks, Self-Organising Maps, Radial Basis Functions, Reinforcement Learning, Hopfield Networks & Boltzmann Machines. Survey of the developments in artificial intelligence, machine learning, expert systems, cognitive science, robotics and artificial neural networks, which contributed to the development of the theory of machine learning systems.
Prerequisites: CS231
Bibliography:

1. C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.
2. S. Haykin, *Neural Networks and Learning Machines*, 3rd Edition, Prentice Hall, 2009.

Teaching methods: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework (laboratory exercises).

CS443 Software Reuse
Type: Restricted Choice
Level: Undergraduate
Semester: Fall or Spring
Credit: 7,5 ECTS units
Instructor: ----------
Objectives: Understanding the usefulness of software reuse. Deepening in the different levels of reuse and understanding the differences between them. Use of software components in practice.
Content: Levels of reuse. Component-based development and composition. Best practices for reuse. Evolution of reuse. Software repositories. Search and retrieval. Design patterns. Object-oriented programming standards. Open source software. Open source licensing and legal issues. Organization policies and open-source based development. Outsourcing. Model-Driven Engineering principles. Service-Oriented Computing. Aspect-Oriented Programming.
Prerequisites: CS233, CS343
Bibliography:

1. M. Ezran, M. Morisio, C. Tully, *Practical Software Reuse*, Practitioner Series, 2002.
2. E. Freeman, E. Robson, B. Bates, K. Sierra, *Head First Design Patterns*, O'Reilly Media, 2004.
3. C. Horstmann, *A Practical Guide to Open Source Licensing*, Wiley, 2nd Edition, 2006.

Teaching methods: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam, homework and semester project.

CS444 Computational Intelligent Systems
Type: Restricted Choice
Level: Undergraduate
Semester: Spring
Credit: 7,5 ECTS units
Instructor: Chr. Christodoulou
Objectives: Global overview of Computational Intelligence and its applications in solving "real" problems in various disciplines such as decision making support, classification, prognosis and prediction, system optimization and recreational design. Moreover there will be an introduction to computational neuroscience/neuroinformatics as well as in cognitive science.
Content: Evolutionary Computing. Genetic Algorithms. Artificial Neural Networks. Fuzzy Systems. Artificial Life. Computational Neuroscience/Neuroinformatics; Hodgkin & Huxley and Integrate-and-Fire neuron models; Neural Coding; Hebbian Learning and Synaptic Plasticity; introduction to cognitive science. Development and Implementation of Computational Intelligence Systems.
Bibliography:

1. P. Engelbrecht, *Computational Intelligence: An Introduction*, John Wiley and Sons, 2nd edition 2007.
2. R. C. Eberhart, Y. Shi, *Computational Intelligence: Concepts to Implementations*, Elsevier, 2007.
3. E. R. Kandel, *Αναζητώντας τη Μνήμη*, (Μετάφραση Α. Καραμανίδης), Πανεπιστημιακές Εκδόσεις Κρήτης, 2008.
4. P. Dayan, L. Abbott, *Theoretical Neuroscience: Computational and Mathematical Modelling of Neural Systems*, MIT Press, 2001.

Teaching methods: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework (laboratory exercises, additional exercises, final project).

CS445 Digital Image Processing
Type: Restricted Choice
Level: Undergraduate
Semester: Fall
Credit: 7,5 ECTS units
Instructor: C. Pattichis
Language: English
Objectives: Introduction to the basic principles of Digital Image Processing: Digital Image and Video. Analysis and implementation of image and video processing and analysis algorithms and their application in industrial and biomedical systems. Content: Binary Image Representation. Image Histogram and Point Operations. Discrete Fourier Transform. Linear Image Filtering. Non Linear Image Filtering Pipeling. Image Compression. Image Analysis I. Image Analysis II. Digital Video Processing.
Prerequisites: CS133, MAS029
Bibliography:

1. R. C. Gonzalez, R. E. Woods, *Digital Image Processing*, 2nd Edition, Addison-Wesley, 2002.

Teaching methods: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework (laboratory exercises, additional exercises, final project).

CS446 Advanced Topics in Databases
Type: Restricted Choice
Level: Undergraduate
Semester: Spring
Credit: 7,5 ECTS units
Instructor: D. Zeinalipour
Objectives: Familiarization with advanced topics in the design and management of Databases (and special kinds of those). Exposure to significant open problems and research directions in the field of Databases.
Content: Theoretical approach to logical and physical design of databases. Algorithms for logical and physical design of databases. Primary and secondary indexing techniques. Advanced query processing and query optimization. Query parallelism. Concurrency control and recovery, integrity and security of data. Distributed databases and introductory concepts distributed transaction processing involving multiple and heterogeneous databases. Problems of interfacing a database with software.
Prerequisites: CS342
Bibliography:

1. R. Ramakrishnan, J. Gehrke, *Database Management Systems*, 3rd Edition, McGraw-Hill, 2003.
2. R. Elmasri, S. B. Navathe, *Fundamentals of Database Systems*, 7th Edition, 2016.

Teaching methods: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam, programming exercises and presentation.

CS447 Computer Vision
Type: Restricted Choice
Level: Undergraduate
Semester: Fall
Credit: 7,5 ECTS units
Instructor: C. Pattichis

Objectives: Familiarity with the basic concepts and methods of developing a simple computerized vision system. Introduction and understanding of commands in a programming language to support the required functionality for developing computational vision systems. Development of applications utilizing the previous.
Content: Basic concepts and methodologies relating to the subject of Computer Vision. Image information, image processing, feature extraction. Image segmentation, clustering, multiple-image processing, case studies.
Prerequisites: CS231, MAS016
Bibliography:

1. D. Forsyth και J. Ponce, Computer Vision: *A Modern Approach*, 2nd Edition, Pearson, 2011.
2. S.D. Prince, *Computer Vision: Models, Learning and Inference*, 2012.
3. R. Hartley, A. Zeisserman, *Multiple View Geometr*y, Cambridge University Press, 2003.
4. C. Bishop, *Pattern Recognition and Machine Learning*, Springer-Verlag, 2007.
5. O. Faugeras, Q.T. Luong, *Geometry of Multiple Images*, MIT Press, 2001.

Teaching methods: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory sessions (1 hour weekly).
Assessment: Final exam, midterm exam, programming exercises and presentation.

CS448 Data Mining on the Web
Type: Restricted Choice
Level: Undergraduate
Semester: Spring
Credit: 7,5 ECTS units
Instructor: G. Pallis
Objectives: Introduction to data mining, Clustering, Classification, Association Rules, Link Analysis, Web communities, Web Personalization.
Content: Data mining on the Web refers to the automatic discovery of interesting and useful patterns from the data associated with the usage, content, and the linkage structure of Web resources. It has quickly become one of the most popular areas in computing and information systems because of its direct applications in e-commerce, information retrieval/filtering, Web personalization, and recommender systems. The primary focus of this course is on examining techniques from data mining to extract useful knowledge from Web data. This course will be focused on a detailed overview of the data mining process and techniques, specifically those that are most relevant to Web mining. Several topics will be covered such as Map-Reduce framework, Web data clustering, classification, association rules, recommendation systems, link analysis, social networks and Web advertising.
Prerequisites: CS231, CS342
Bibliography:

1. A. Rajaraman, J. D. Ullman, *Mining of Massive Datasets*, Cambridge University Press, 2011.
2. J. Han, M. Kamber, *Data Mining: Concepts and Techniques*, 2nd Edition, Morgan Kaufmann, 2006.
3. B. Liu, *Web Data Mining: Exploring Hyperlinks, Contents and Usage Data*, Springer, 2007.

Teaching methods: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam, homework (programming and theoretical assignments) and semester project.

CS449 Professional Practice in Software Engineering
Type:Restricted Choice
Level: Undergraduate
Semester: Spring
Credit: 7,5 ECTS units
Instructor: G. Kapitsaki

Language: English
Objectives: Embedding and practical application of the theoretical approaches and methodologies of Software Engineering for the development of a product-software system that serves the needs of an organization belonging to the local market. Practical use of processes and tools, such as central and distributed version control systems (SVN and GitHub), testing at different levels, group communication, professionalism, and ethical conduct.
Content: Undertake and carrying out to completion a significant software project by small student groups (of about 2-6 students each). All phases in the development of software. Some of the specific projects come from the industrial sector. Version control systems (Git and GitHub). Use of the Scrum development methodology (Scrum review meeting, Scrum retrospective, etc.). Testing and automated testing. Software system analysis through software metrics. Specialized issues depending on the project nature (e.g., deployment on web servers, GUI tools and frameworks etc.).
Prerequisites: CS343
Bibliography:

1. Selected articles

Teaching methods: Students are grouped in teams of 2-6 persons. Meetings/discussions are held regularly (weekly per team), Recitation (1 hour weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Assessment of the product-software system, assessment of the corresponding documentation, quizzes, oral presentation and final exam.

CS450 Network Virtualisation and Management
Type: Restricted Choice
Level: Undergraduate
Semester: Fall
Credit: 7,5 ECTS units
Instructor: A. Pitsillides / V. Vassiliou
Language: English
Objectives: Familiarization with fundamental topics in the management of communication networks, with a focus on the Internet and latest trends, including SDN, OpenFlow, and Network Virtualisation.
Content: Network management is an essential component in today's networks, as the world increasingly relies on networked systems and services to function. The course aims to cover the latest approaches in network management, including virtualisation of the network itself (e.g. Software Defined Networks, Network Function Virtualisation, Virtual Infrastructure Management, VNF-Virtual Network Functions, NFV Management and orchestration, Network Services) and the cloud (e.g. OpenStack), as well as more traditional techniques of network management (e.g. SNMP).
Prerequisites: CS324
Bibliography:

1. M. Subramanian, *Network Management: Principles and Practices*, 2nd Edition, 2012, Prentice Hall.
2. A. Clemm, *Network Management Fundamentals*, 2006, Cisco Press® Fundamentals Series.
3. N. L. S. da Fonseca, R. Boutaba, *Cloud Services, Networking, and Management*, IEEE Press Series on Networks and Services Management, April 2015.

Teaching methods: The course combines lectures and tutorials with assignments, labs and experimentation with networking management techniques and tools. It will be delivered in the form of Seminars, which will cover theory and be intertwined with hands-on experience in a real context, where students will apply theory, techniques and available tools in a real network. Seminar type Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Individual project, assignments and laboratory work. No exams will be given.

CS451 Software Analysis
Type: Restricted Choice

Level: Undergraduate
Semester: Fall
Credit: 7,5 ECTS units
Instructor: E. Athanasopoulos
Language: Greek or English
Objectives: The course explores fundamental concepts in analyzing software of multiple forms and for different purposes. Many times, we need to analyze software for (a) locating bugs (debugging), (b) measuring performance bottlenecks (profiling), (c) adding instrumentation that enhances a program's behavior (e.g., add a security defense). The course exposes several techniques for working directly with the binary form of software (binary analysis and re-writing), as well as exploring and augmenting the source of C/C++ programs through the extension of modern compiler toolchains (LLVM).
Content: ELF format of Unix binaries. Tools that can work and explore binaries in Unix (show different sections, symbols, shared libraries, etc.). How relocations and shared libraries work in binaries (e.g., the usage of GOT). Using ptrace(). Disassembling binaries using the Capstone framework. Re-writing binaries programmatically. Pre-loading binaries. Dynamic and static analysis of binary code. C/C++ instrumentation through LLVM passes. Applications of software analysis.
Prerequisites: CS211, CS232
Bibliography:

1. Practical Binary Analysis: Build Your Own Linux Tools for Binary Instrumentation, Analysis, and Disassembly. Dennis Andriesse. ISBN-10: 1593279124.
2. LLVM documentation
3. Published papers.

Teaching methods: Lectures (3 hours weekly) and hands-on sessions (1 hour weekly).
Assessment: Final exam, midterm exam, programming assignments.

CS481 Software Engineering for Software as a Service
Type: Restricted Choice
Level: Undergraduate
Semester: ……
Credit: 7,5 ECTS units
Instructor: G. Kapitsaki
Objectives: Understanding of advanced concepts of software engineering. Use of Agile software methodology with applications for cloud computing. Practice via the implementation of a software system. Regression testing.
Content: Agile development models. Versoion control systems. Software systems for Software as a Service (SaS). Test-driven development, user-centric design, refactoring and deployment. License compliance. Pair programming.
Prerequisites: CS343
Bibliography:

1. A. Fox, D. Patterson, *Engineering Software as a Service: An Agile Approach Using Cloud Computing*, 2nd Edition, 2.0b7, 2021.
2. Selected articles from the literature.

Teaching methods: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam, homework and semester project.

CS482 Software Validation, Verification and Quality
Type: Restricted Choice
Level: Undergraduate
Semester: ……
Credit: 7,5 ECTS units
Instructor: G. Kapitsaki / G. Papadopulos
Objectives: Teaching and assimilation of basic and specialized concepts on software validation and verification. Acquisition of knowledge for designing and implementing different types of audits and debugging procedures. Deepening in concepts and practices for

ensuring quality in software systems.
Content: Basic concepts and terminology for validation and verification, reviews, inspections, control (parts, system combination, regression, acceptance, coverage criteria), control of specific categories of software systems, analyzing problems and documentation. Basic principles of quality factors and quality characteristics of software, methodologies, tools, quality procedures, quality standards, models and quality metrics, quality software production process, quality plan, organize quality assurance process, quality assessment process.
Prerequisites: CS343
Bibliography:

1. P. Jorgensen, *Software Testing, A Craftsman's Approach*, 3rd edition, Auerbach Publications, 2008.
2. P. Ammann and J. Offutt, *Introduction to Software Testing*, Cambridge University Press, 2008.
3. J. Tian, *Software Quality Engineering: Testing, Quality Assurance, and Quantifiable Improvemen*t, Wiley, 2005.
4. William E. Lewis, Gunasekaran Veerapillai, *Software Testing and Continuous Quality Improvement*, 3rd Edition, 2008.

Teaching methods: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam, homework and laboratory assignments.

CS 483 Software Analysis, Modelling and Design
Type: Restricted Choice
Level: Undergraduate
Semester: ……
Credit: 7,5 ECTS units
Instructor: G. Kapitsaki / G. Papadopulos
Objectives: Deepening in fundamental concepts of software analysis, modelling and design. Design of effective, efficient and competitive software products. Training in software architectures, notations, standards and methods.
Content: Fundamental concepts of software modeling (principles, conditions, properties), modeling languages, virtual models, model types. Early analysis (consistency, sufficiency, accuracy, quality). The role of models in the development and the connection with software development methodologies. Model-driven engineering. The role of software architecture in the software development lifecycle (user needs, design, implementation). The architectural design of a software system. Architectural representation.
Prerequisites: CS343
Bibliography:

1. K. Qian, X. Fu, L. Tao, C. Xu, *Software Architecture and Design Illuminated*, Jones and Bartlett Learning, 2010.
2. A. Kleppe, J. Warmer, W. Bast, *MDA Explained. The Model Driven Architecture: Practice and Promise*, Addison-Wesley Professional, 2003.
3. R.N. Taylor, N. Medvidovic and E.M. Dashofy, *Software Architecture*, Wiley, 2010.

Teaching methods: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam, homework and laboratory assignments.

CS 499 Special Issues in Computer Science
Type: Restricted Choice
Level: Undergraduate
Semester:----------
Credit: 7,5 ECTS units
Instructor: .....
Objectives: The objective of this course is to promote the technology culture in the undergraduate program of the Computer Science Department. The course focuses on a specific topic of Computer Science. Further information regarding the prerequisites,

description, requirements and assessment of the course will be presented when the course is announced on the Department's website.
Content: The content of the course is formulated according to the specific topic.
Prerequisites: The consent of the instructor.
Bibliography: ----------
Teaching methods: Lectures (3 hours weekly) and Recitation (1 hour weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework (laboratory exercises and additional exercises).

CS 500 Industrial Placement
Type: Restricted Choice
Level: Undergraduate
Semester: Summer
Credit: 7,5 ECTS units
Instructor: E. Athanasopoulos / Chr. Georgiou / M. Dikaiakos
Objectives: Linking and applying in a real work environment the knowledge that students have acquired in the curriculum courses. Bringing students in direct contact with the new trends and needs of the industrial market as well as the demand for specific skills and competencies. Providing the students with a first experience and opportunity to adapt to a working environment and to recognize the requirements of a professional space as well as the possibility of demystifying some fears about this space: all necessary for the preparation for future entry into the industry market. In exceptional cases, it may be a precursor to future professional cooperation between the student and the employer.
Content: The industrial placement provides students with employment for a short period of time in a real work environment and on a subject related to the curriculum of the Department of Computer Science.
Prerequisites: Completion of the 3rd year of study (completion of at least156 ECTS).
Bibliography: ----------
Teaching methods: ----------
Assessment: Final report.

Courses from Department of Business and Public Administration

BPA 369 Principles of Entrepreneurship and Innovation
Type: Compulsory
Level: Undergraduate
Semester:----------
Credit: 7,5 ECTS units
Instructor: Daina Nicolaou
Objectives: To explain the process of innovation and entrepreneurship (inter-dependent concepts) in an understandable and simple way. Additionally, students will be guided through identifying a business opportunity to setting up and growing a business (including how to perform a successful exit). The lectures are based on academic theory, but the emphasis is focused on the practical application of this theory. To familiarize students with theory and practice about entrepreneurship and the management of new businesses while simultaneously emphasizing the role played by new business ideas for the economy.
Content: Key Partners. Key activities. Key resources. Value creation. Customer Segments. Communication/Distribution channels. Customer Relationships. Cost structure and Expenses. Revenue streams. Additional topics: business plans, venture capital firms, angel investors, intellectual property protection, sources and diffusion of innovation.
Prerequisites: --------
Bibliography:
1. N. M. Scarborough, Ξ. P. Cornwall, *Essentials of Entrepreneurship and Small Business Management*, Pearson, 2019.

2. Harvard Business School Case Pack.

Teaching methods: Lectures, talks by entrepreneurs, visits to companies, case studies, project preparation.
Assessment: Final exam , group assignment, activities (e.g. business model presentation, pitching), attendance and participation.

Courses for Other Departments

These courses are offered to students of other Departments. The content of such courses is suitably determined so that students in other disciplines may appreciate the significance of Computer Science, its relationship to other disciplines, and the potential benefits it offers. Each of the courses for other Departments carries 5, 6 or 7 ECTS units. The courses may be offered every semester or in parallel classes, depending on the needs and capabilities.

CS001 Introduction to Computer Science
Type: Compulsory for the students of the departments of PSY and SPS
Unrestricted Choice (for students of other departments)
Level: Undergraduate
Semester: Fall and Spring
Credit: 6 ECTS units
Instructor: ……. (Fall) / …….. (Spring)
Objectives: Introduction to the basic concepts and the wide range of Computer Science. Familiarization and global update of students with the structure and use of computers, computer programs, the Internet and the application of Computer Science to other fields.
Content: The course aims to provide students with basic understanding of computer science concepts. The objective of the course is to introduce the students to the 'modern' research trends of the computer science field and the various applications of computer science in other areas. Furthermore, to allow students to appreciate the potentials of informatics and especially the web in their working environment. Students will also become aware of internet safety and malicious software and how they can protect themselves and their data at personal and professional level. Through the practical laboratory sessions, students will get familiar with various tools and software that is considered vital for their academic and professional career. Specifically, the course will cover the following topics: Internet safety and malicious software, social networking, introduction to Web 2.0, search engines, social networking, bibliography and citation management tools.
Prerequisites: ---------
Bibliography: ---------
Teaching methods: Lectures (3 hours weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework (laboratory assignments).

CS002 Introduction to Computer Science
Type: Compulsory for the students of the departments of EDU
Level: Undergraduate
Semester: Fall and Spring
Credit: 5 ECTS units
Instructor: …….
Objectives: Introduction to the basic concepts and the wide range of Computer Science. Familiarization and global update of students with the structure and use of computers, computer programs, the Internet and the application of Computer Science to other fields.
Content: Fundamentals of Computer Science, the main historical events which have contributed in its development, and the possibilities it offers. Basic constituent elements of Computer Science and methods for making it valuable to other sciences and applications. Practical experience with application packages, and the UNIX environment. Basic principles of programming in a fourth generation language.
Prerequisites: ----------
Bibliography:

1. B. A. Forouzan, *Εισαγωγή στην Επιστήμη των Υπολογιστών*, Εκδόσεις "ΚΛΕΙΔΑΡΙΘΜΟΣ", 2003.

Teaching methods: Lectures (3 hours weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework (laboratory assignments).

CS003 Computer Science and Information Systems
Type: Compulsory for the students of the departments of ECO and PBA
Level: Undergraduate
Semester: Fall and Spring
Credit: 6 ECTS units
Instructor: …… (Fall) / …… (Spring)
Objectives: Familiarization with the most basic concepts in Computer Science, Information Systems and Computer Systems. Touch with the current trends in the practice of Computer Science. Practical experience in the use of various software packages that are useful in the academic and professional worlds.
Content: The course aims to provide students with basic understanding of computer science concepts. The objective of the course is to introduce the students to the 'modern' research trends of the computer science field and the various applications of computer science in other areas. Furthermore, to allow students to appreciate the potentials of informatics and especially the web in their working environment. Students will also become aware of internet safety and malicious software and how they can protect themselves and their data at personal and professional level. Through the practical laboratory sessions, students will get familiar with various tools and software that is considered vital for their academic and professional career. Specifically, the course will cover the following topics: Internet safety and malicious software, social networking, introduction to Web 2.0, search engines, introduction to cloud computing , introduction to databases, introduction to web design, advanced excel (excel statistics and economical formulas), data visualisation, social networking for commercial and marketing purposes.
Prerequisites: ----------
Bibliography: ---------
Teaching methods: Lectures (3 hours weekly) and Recitation/Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework.

CS011 Introduction to Information Society
Type: Unrestricted Choice (for students of other departments)
Level: Undergraduate
Semester: …….
Credit: 6 ECTS units
Instructor:…….
Objectives: Familiarization with the most prominent concepts and applications of Information Society.
Content: Presentation of the formed framework for Information Society (IST). Basic concepts and constituent elements of IST, and the wider context for its application. Issues such as electronic government, telematics, digital business, electronic commerce, telemedicine, etc. Effects of IST on society and economy.
Prerequisites: ---------
Bibliography:

1. B. A. Forouzan, *Εισαγωγή στην Επιστήμη των Υπολογιστών*, Εκδόσεις "ΚΛΕΙΔΑΡΙΘΜΟΣ", 2003.

Teaching methods: Lectures (3 hours weekly).
Assessment: Final exam and homework (individual midterm project and final group project).

CS012 Web Design Technologies
Type: Unrestricted Choice (for students of other departments)
Level: Undergraduate
Semester: Spring
Credit: 6 ECTS units

Instructor:…….
Objectives: (a) basic concepts of the Internet and the WWW, (b) well-known technologies for designing and developing websites and (c) the use of specialized web development software.
Content: Introduction to the Internet and the WWW. Web design and development technologies. Web servers and HTTP, HTML, XHTML, CSS, Javascript. User interface design guidelines. Usability evaluation.
Prerequisites: --------
Bibliography:

1. Σ. Ρετάλης Γ. Τσέλιος, HTML: *Μαθήματα από το Απλό στο Σύνθετο*, Εκδόσεις Καστανιώτη, 2003.

Teaching methods: Lectures (3 hours weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework.

CS013 History and Philosophy of Computation
Type: Unrestricted Choice (for students of other departments)
Level: Undergraduate
Semester: ----------
Credit: 6 ECTS units
Instructor: A. Kakas
Objectives: Basic understanding of the foundational motion of computation, Computational machine and intelligence through their historical evolution. Theoretical and practical understanding of the automation of computation. Understanding of the relationship between computing and human intelligence.
Content: Foundational motions of computation, complexity, computational machine, algorithm, programming language, knowledge and intelligence. Basic computational models and their corresponding problems. Historical development of computers and their theoretical basis. The scientific and engineering character of computer science. The relation of computer science to other disciplines such as mathematics, physics, psychology and biology. Computing today over the internet and its future development. Artificial intelligence and machines with intelligent behavior. The human brain and mind as computing machines. Automation of logic and argumentation. Natural language as a computing language.
Prerequisites: --------
Bibliography:

1. L. Floridi, *Guide to the Philosophy of Computing and Information*, Backwell 2004.
2. Selected articles from the Journal of Minds and Machines.

Teaching methods: Lectures (3 hours weekly) and Recitation (1 hour weekly).
Assessment: Final exam and homework.

CS014 Aristotle, Logic and Ethical Computer Science
Type: Unrestricted Choice (for students of other departments)
Level: Undergraduate
Semester: Spring
Credit: 6 ECTS units
Instructor: A. Kakas
Objectives: Understanding the evolution Understanding the evolution of Logic from ancient times through a modern understanding of the needs of Information Technology and especially Artificial Intelligence. Understand the relationship between Aristotle's Logic with ethical values and the ethics of Artificial Intelligence machines.
Content: Fundamental concepts of reasoning and logic by Aristotle. Distinguishing between evidence and dialectical reasoning (Aristotle). Linking the two types of reasoning with the needs of IT yesterday and today. From Aristotelian logic to Boolean logic and Digital logic. Dialectical Argumentation by Aristotle to the Present and its Relation to Computational Argumentation in Artificial Intelligence. Aristotle's Collective as a Logic and Computational Framework. Strict reasoning and reasoning of man. Reasoning Automation and Cognitive Computing Arguments. Relation of Aristotle's Logic to Ethical Values and the Ethics of Artificial Intelligence Machines. Practical reasoning and framework for ethical behavior of

people and machines.
Prerequisites: --------
Bibliography:

1. Aristotle, *Όργανον*, and *Ηθικά Νικομάχεια*.
2. H. White, M. Shenefelt: *If A Then B: How the World Discovered logic*, Columbia University Press, 2013.
3. EE AI HLEG, *Ευρωπαϊκός Κώδικας «Ηθικής και Αξιόπιστης Τεχνητής Νοημοσύνης»*, 2019.
4. B.J. Gross et al., *Embedded EthiCS: Integrating Ethics Across CS Education*, Communications of the ACM, Vol62 (No 8), pp 54-61, 2019.

Teaching methods: Lectures (3 hours weekly).
Assessment: Final exam (50%) and assignments (50%).

CS031 Introduction to Programming
Type: Compulsory for students of MAS Department
Level: Undergraduate
Semester: Spring
Credit: 7 ECTS units
Instructor: …….
Objectives: Introduction of the basic principles of programming with emphasis on structured programming, abstraction, and the design, implementation, checking and debugging of modular programs. Application of these principles using the FORTRAN 90/95 programming language.
Content: Computers and binary system. Hardware and software. Program development cycle, algorithms and flow diagrams. Alphabet and syntax of FORTRAN. Operators. Selection structures and loops. Arrays. Functions and subroutines. Recursion. Formatted input-output. Files. Dynamic data.
Prerequisites: ----------
Bibliography:

1. Δ. Ματαράς, Φ. Κουτελιέρης, *Προγραμματισμός Fortran 90/95 για Επιστήμονες και Μηχανικούς*, Εκδόσεις ΤΖΙΟΛΑ, 2003.
2. L. Nyhoff, S. Leestma, *Introduction to Fortran 90*, Prentice Hall, 1999.

Teaching methods: Lectures (3 hours weekly), Laboratory sessions (1.5 hours weekly) and Recitation (1 hour weekly).
Assessment: Final exam, midterm exam, homework (theoretical and programming assignments) and quizzes.

CS032 Introduction to Computer Science & Information Systems
Type: Compulsory (for the students of the departments of ECO and PBA)
Level: Undergraduate
Semester: Spring
Credit: 6 ECTS units
Instructor: …….
Objectives: Programming is examined as a problem-solving method. In particular the course presents the fundamentals of algorithmic thought and the implementation thereof through a programming language. Also, a high level programming language is introduced. Upon completion of the course students are expected to be able to cast problem solutions into an algorithmic form, and will have obtained a basic exposure to a widely used programming language such as C or Python.
Content: Introduction to the principles of programming with emphasis on structured programming, abstraction, and the design, implementation, checking and debugging of modular programs. Mastering the material through laboratory exercises in the C programming language.
Prerequisites: --------
Bibliography:

1. W. F. Punch, R. Enbody, The Practice of Computing Using Python, 2nd Edition, Addison-Wesley, 2013..

Teaching methods: Lectures (3 hours weekly) and Recitation/Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework.

CS033 Introduction to Programming for Engineers
Type: Compulsory for students of the Department of CEE
Level: Undergraduate
Semester: Fall
Credit: 5 ECTS units
Instructor: …….
Objectives: Introduction of methods for problem-solving through programming. Provision of deep understanding of basic programming principles and algorithmic techniques, design, implementation, testing and debugging of modular programs. Application of the basic principles using the C programming language.
Content: Basic principles of programming with emphasis on structured programming, abstraction, and the design, implementation, checking and debugging of modular programs. Mastering of the material through laboratory exercises in a traditional programming language such as C.
Prerequisites:---------
Bibliography:

1. J. R. Hanly, E. B. Koffman, *C Program Design for Engineers*, Second Edition, Addison-Welsey, 2001.

Teaching methods: Lectures (3 hours weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exams and homework (including programming assignments).

CS034 Introduction to Programming for Electrical and Computer Engineers
Type: Compulsory for students of the Department of ECE
Level: Undergraduate
Semester: Spring
Credit: 7 ECTS units
Instructor: …….
Objectives: Introduction of methods for problem-solving through programming. Development of procedural problem solving skills and algorithmic thinking. Provision of deep understanding of basic programming principles and algorithmic techniques, design, implementation, testing and debugging of modular programs. Understanding the important concepts such as program abstraction and data abstraction. Mastering of a high-level programming language (C).
Content: Introduction to computers and programming languages. Problem solving and programming, problem specification, algorithms and programs, modular programming, program and data abstraction. Software development process, top-down design, problem decomposition, reuse, trial and debugging. Variables: names, values, addresses, basic types (numbers, characters, logical values), operators and expressions, constants, library usage. Input/Output operations. Procedures (functions), parameters, calls, value or address referral. Program flow, variables' scope, lifecycle of variables/function calls, program's state. Procedural programming, algorithmic structures (sequence, selection, loop, recursion), memory. Synthesized and enumerated data types, arrays (vectors and multidimensional), structures, pointers (variables of pointer type, address and indirect referral operators, arrays and pointers and functions). Introduction to dynamic memory allocation.
Prerequisites:---------
Bibliography:

1. J. R. Hanly, E. B. Koffman, *Problem Solving and Program Design in C*, Fourth Edition, Addison-Wesley, 2003.
2. B. W. Kernighan, D. M. Ritchie, *Η Γλώσσα Προγραμματισμού C*, 2nd Edition, Εκδόσεις ΚΛΕΙΔΑΡΙΘΜΟΣ, 1990.

Teaching methods: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exams and homework (programming assignments) and quizzes.

CS035 Data Structures and Algorithms for Electrical and Computer Engineers
Type: Compulsory for students of the Department of ECE
Level: Undergraduate
Semester: Fall
Credit: 7 ECTS units
Instructor: …….
Objectives: This course studies methods of efficient organization and manipulation of data, complexity analysis of algorithm design techniques. The main objectives of the course are to familiarize students with data structures and their associated algorithms, techniques for evaluating the complexity of algorithms, and the development of skills for efficient algorithm design and implementation.
Content: Advanced programming techniques based on the programming language C: Recursion, Structures, Pointers, File and Memory management. Data types and abstract data types. Algorithm complexity analysis: worst-case and average-case analysis. Linear data structures: List, Stack and Queue, using static and dynamic memory allocation methods. Applications of linear data structures. Sorting algorithms: SelectionSort, InsertionSort, MergeSort, QuickSort and BucketSort. Tree data structures: Binary Trees, Binary Search Trees, Balanced Trees, B-trees. Priority Queues and Heaps. Graphs: definitions, data structures, topological sorting algorithms, graph traversal algorithms. Hashing techniques, hash functions and collision resolution techniques.
Prerequisites: CS034
Bibliography:

1. R. F. Gilberg, B. A. Fourouzan, *Data Structures: A Pseudocode Approach with C*, 2nd Edition, Thomson Publishing, 2005.
2. K. N. King, *C Programming: A Modern Approach*, 2nd Edition, W. W. Norton & Company, 2008.
3. Ν. Μισυρλής, *Δομές Δεδομένων με C*, 2002.
4. M. Allen Weiss, *Data Structures and Algorithm Analysis in C,* Addison Wesley, 1996.

Teaching methods: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework (theoretical and programming assignments)

CS041 e-Health and Medical Informatics
Type: Compulsory for students of the Medical School
Level: Undergraduate
Semester: Fall
Credit: 6 ECTS units
Instructor: ----------
Objectives: Introducing the doctor of the future to the new world order of electronic health (e-health) and medical informatics at local, European and international level. Consolidating the legislative and social framework of e-health. Explaining and appreciating the patient-centric approach to medical practice as a prerequisite for successful implementation of e-health. Exploitation of the potential offered by information and communication technologies in medicine and clinical practice, particularly through the modeling of medical practice, procedures and knowledge. Appreciating the importance of the available medical data for making e-health a reality.
Content: Introduction to the e-health environment and the appropriate operating framework. Legislative regulatory and social background needed for its materialization. The importance of information technology in extracting useful information from vast medical databases. Applications of computer systems used for the movement of medical knowledge, medical

information management, proper use of a citizen's electronic folder for patients and support a medical decision. Reference to the legal framework that regulates the medical practice, in accordance with European and international directives.
Prerequisite: ---------
Bibliography: no book that covers all material expected to be taught in the course exists. Extensive use of the Internet will be recommended, even in the classroom. The following books are recommended.

1. D. Koutsouris, S. Pavlopoulos, A. Prentza, *Introduction to Biomedical Engineering and Systems Analysis*, Publisher: Tziola, 2003.
2. E. H. Shortliffe, J. J. Cimino (Eds.), *Biomedical Informatics: Computer Applications in Health Care and Biomedicine*, Springer Verlag; 3rd edition, 2006.

Teaching methods: Lectures/presentations (3 hours weekly) and Recitation (1 hour weekly), discussions/presentations. Structured visits to hospital clinics for data collection and exploitation as to how these can be used by the medical staff.
Assessment: Written exams (midterm and final), essays (studies and/or exercises).

CS042 eHealth Seminars
Type: Compulsory for students of the Medical School
Level: Undergraduate
Semester: Spring
Credit: 2 ECTS units
Instructor: ----------
Objectives: Familiarization with the information and communication technologies (ICT) and their practical application in medicine and the medical environment. Acquiring of knowledge through eHealth practices adopted in real life by the distinguished presenters of the seminars. Study through videos practices in real environments such as, the intensive care unit, laparoscopic operations, robotic assisted operations, telediagnosis, teleconsultation; appreciation of the importance of medical databases. Understanding of what is coming up in future medicine, considering that ICT will continue to grow and offer its services to the medical profession.
Content: Selected presentations/demonstrations by at least six medical experts selected from Cyprus or abroad according to their availability. Teleconferencing may be used for real life presentations and communication with the selected presenters.
Prerequisite: CS041
Bibliography: no book that covers all material expected to be taught in the course exists. Relevant bibliography will be provided by the presenters of the seminars and the Instructor of the course.
Teaching methods: Lectures/presentations (2 hours biweekly). Structured visits to hospital clinics in coordination with the presenters and the Medical School.
Assessment: Written e-exam, assignments (one individual and one group assignment).

Post Graduate Program of Studies

The postgraduate programme of studies leads to Master and Ph.D. Degrees in Computer Science.

The postgraduate studies in the Department are governed by Corresponding Rules of Post Graduate Studies, approved by the Senate (149th meeting 22/5/2002). These rules are included in the Appendix B of this guide. Also in Appendices C and D present the specifications to be met by the final form of the Master and Ph.D. dissertations, respectively.

Masters Programs

The Department of Computer Science, since the academic year 2009-2010, offers the following four Master Programmes:

1. Master in Computer Science (MCS)
2. Professional Master in Advanced Information Technologies (PM).

Since 2017/2018 the Department offers in collaboration with the Open University of Cyprus and the Department of Psychology of the University of Cyprus the following Master Programme:

3. Master Program:Master in Cognitive Systems

Since 2021/2022 the Department offers in collaboration with the Department of Business and Public Administration and the Department of Mathematics and Statistics the following Master Programme:

4. Master in Data Science

Since 2022/2023, the Department offers, in English the master developed in the context of project MAI4CAREU, co-funded by Europe, and having an international network of other European Universities, the CYENS Centre of Excellence and several high-tech SMEs:

5. Master in Artificial Intelligence

In accordance with the ECTS System, successful completion of at least 90 ECTS is required for obtaining a Master Degree at the postgraduate level.

To be admitted to a Master programme, a candidate must have a first degree in Computer Science or a related subject from an accredited university with an overall grade of "Very Good". Any relevant industrial experience may be considered as an additional advantage.

In the Master Programs that include a Master Thesis, the thesis should be submitted upon completion to the digital library of the University of Cyprus, Lekythos (https://lekythos.library.ucy.ac.cy/?locale-attribute=en). A brief description of these four masters:

1. MASTER in COMPUTER SCIENCE

The Master Programme in Computer Science is designed primarily for Computer Science and other science-related graduates who seek to develop research skills and enhance their knowledge in advanced areas of Computer Science. Students who attend this programme may pursue a PhD degree after their graduation.

The completion of the programme requires 90 ECTS, and the duration of studies should be

at least three semesters. These 90 credits correspond to eight courses and a Master thesis. More specifically:

- Seven postgraduate courses of 8 ECTS each (any seven from the postgraduate course list with the supervision of the Academic Advisor)
- One postgraduate course of 4 ECTS (CS670 Research Methodologies and Professional Practices in Computer Science)
- Master Thesis worth of 30 ECTS.

The seven graduate courses must be from the graduate course catalogue of the Department. Each student selects these courses with the guidance of the academic advisor.

In case a student successfully complete at least four courses from the specialized Master Direction of the Department then the student may transferred to the corresponding Specialzed Master where will receive his diploma.

2. MASTER in ADVANCED INFORMATION TECHNOLOGIES (PROFESSIONAL)

The aim of the Professional Master in Advanced Information Technologies is to help Information Technology professionals to extend and update their knowledge in Advanced Computer Technologies and to acquire up-to-date know-how in subjects related to the national Information Technology industry like Software Engineering, the Internet, and Intelligent Systems.

The completion of the course requires 90 ECTS and the duration of studies must be at least four semesters. In particular:

- Nine postgraduate courses of 8 ECTS, out of which 4 should be related to the programme´s scope (identified as such in the Table of Specialization Courses and in the course descriptions)
- One postgraduate course of 4 ECTS (EPL 670 Research Methodologies and Professional Practices in Computer Science)
- Master Thesis worth of 15 ECTS, which can be replaced with two extra postgraduate courses.

The needs of employees and professionals in the Information Technology industry will be considered during the scheduling of courses (afternoon and evening courses and three-hour meetings).

3. MASTER in COGNITIVE SYSTEMS

This is a distance-learning program taught exclusively on-line in collaboration with the Open University of Cyprus and the Department of Psychology of the University of Cyprus. The program is taught in English.

Cognitive systems form a new generation of systems that aim to collaborate with their users at a level cognitively compatible with a non-computing expert in order to provide personalized and adaptive services, with each party, system and human, learning and adapting to the capabilities of the other. The need for developing such cognitive systems has been widely recognized. The Watson machine opened up the area of Cognitive Computing and we see today the development of the first Cognitive Assistants such as Siri, Cortana, Alexa, Google Assistant and others from every major computing company in the world.

Students are required to take:

- 3 first courses under the Foundations theme (COS511, COS512, COS513), where at least the two introductory courses (COS511, COS512) are expected to be taken during the first semester.
- electives among all other courses, as long as at least one third of the courses comes from Cognitive Psychology (CP), and one third of the courses come from Computer Science (CS).

Full information about the program and how to apply can be found at: http://cogsys.ouc.ac.cy

4. MASTER in DATA SCIENCE (MDS)

The programme is offered jointly by the Department of Business and Public Administration, the Department of Computer Science, and the Department of Mathematics and Statistics since September 2021 and it is offered in English. Data science is an inter-disciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from Big Data. Data Science is quickly becoming a field of central importance to the strategy of modern organizations. There is an increasing need for highly trained employees who can think across disciplines to transform data into actionable insights. The objective of the program is to provide students with a strong understanding of basic and advanced methods in statistical inference, machine learning, data visualization, and data mining, which are the essential skills a modern data scientist needs to posses.

The completion of the programme requires 90 ECTS and the duration is 1,5-year. The programme offers 3 tracks (Computer Science Track / Statistics Track / Business Analytics Track). The first two semesters will be dedicated to core courses, while students will select a track at the end of the second semester.

Full information about the program, as well as the descriptions of the offered courses, can be found at: https://datascience.cy/.

5. MASTER in ARTIFICIAL INTELLIGENCE (MAI)

The aim of the new MSc Artificial Intelligence programme, starting in September 2022, and which will be delivered in English, is to be a modern programme, containing a strong interdisciplinary element as required by human-centric, explainable, and responsible artificial intelligence. Its compulsory courses include courses on artificial intelligence and ethics, as well as on artificial intelligence and entrepreneurship. Providing career counselling to students is a high priority, with the aim of helping all graduates to successfully pursue an AI-related career, possibly set up their own start-ups.

The development of this new master's programme is co-funded by the European Union (Connecting Europe Facility (CEF) — Telecommunications Sector). Europe's initiative to fund new master's programmes in Artificial Intelligence demonstrates the importance Europe attaches to recent developments in this area, in providing solutions to global problems related to every aspect of human life, contributing to growth and competitiveness. Europe urgently needs more professionals in Artificial Intelligence, entrepreneurs, and researchers, capable of paving the way for new innovations for the good of society and pushing the borders of the field towards new challenges. This new generation of AI graduates must be fully familiar with the latest technological developments in terms of the breadth and depth of technical knowledge in Artificial Intelligence, be connected to industry and be fully aware of the ethical issues involved.

The programme's collaboration network includes four European Universities (University of Cyprus, University of Bologna, Technische Universität Dresden, and University of Ruse

Angel Kanchev), the CYENS Centre of Excellence and four high-tech SMEs (Nodes & Links Labs, 3AeHealth LTD, INJENIA SRL, MLPS AD). Fostering strong links with industry provides significant added value to the educational experience.

The programme's intended learning outcomes are the following:

1. Grasp the fundamental principles and techniques that underline software systems that exhibit "intelligent" behavior, be aware of the latest developments in AI and appreciate the wide breath of study that AI requires.
2. Acquire a holistic view of machine learning enabling them to understand the principles that drive most scientific and industrial AI innovations.
3. Understand how to provide computers with the ability to intelligently process human language, extracting meaning, information, and structure from text, speech, web pages, and social networks.
4. Master key concepts and challenges of relevance to AI and data-driven entrepreneurship.
5. Be able to apply methods and tools of AI research, and be aware of professional practices, and associated technological culture, bearing in mind EC's regulatory framework.
6. Understand the basics of implementing systems that are not only highly performing, but also adhere to our ethical socio-legal cultural values and to understand how research into AI ethics feeds into policy and how policy requirements affect the development of AI systems.
7. Acquire deep knowledge and familiarity with current and future research challenges in several specialized topics covered in the elective courses.
8. Be aware of the career opportunities in the various countries.
9. Exhibit versatility and innovative thinking in addressing and managing open questions in a variety of contexts, as an essential asset for careers in AI.
10. Develop transferable skills such as: oral and written scientific communication, near fluent use of scientific English, use of information/communication technology, organization and planning of group work.

Program Structure

Semester/Courses	ECTS Credits
Fall Semester 1	30
MAI611 AI Fundamentals	8
MAI612 Machine Learning	8
MAI613 Research Methodologies and Professional Practices in AI	4
MAI614 AI on the Edge Webinars I	2
Elective course 1	8
Spring Semester 1	30
MAI621 AI Ethics I	6
MAI622 AI Entrepreneurship	8
Elective course 2	8
Elective course 3	8
Summer Period 1 (Student participation is optional)	12
MAI601 AI Camp	4
MAI602 Research/industrial internship (On the basis of evaluation an internship could count as an elective)	8
Fall Semester 2	30
MAI631 AI Ethics II and AI Policy Making	4
MAI632 AI on the Edge Webinars II	2
Elective course 4	8
MAI641 Master Thesis OR Elective course 5 and Elective course	16
MSc Total ECTS	90-102

Elective Courses (8 ECTS each):

MAI623 Natural Language Processing
MAI642 Deep Learning
MAI643 AI in Medicine
MAI644 Computer Vision
MAI645 Machine Learning for Graphics and Computer Vision
MAI646 Cognitive Programming for Human-Centric AI
MAI647 Computational Neuroscience
MAI648 Human-Centered Intelligent User Interfaces
MAI649 Principles of Ontological Databases
MAI650 Internet of Things
MAI651 AI and Creativity

Additional Elective Courses from Master in Data Science:
DSC511 Data Analytics
DSC551 Data Visualization
DSC516 Cloud Computing

Information about the program can be found at: https://www.cs.ucy.ac.cy/index.php/education/postgrad/master-in-artificial-intelligence and https://mai4car.eu.

Table 2: Specialization Courses in Graduate Programs

Course Code and Title	Master in Computer Science[1]	Professional Master	Semester 2020-21
CS601 – Distributed Systems	√		Winter
CS603 – Using Software Architectures for the Design and Development of Software Systems	√	√	Winter
CS604 – Artificial Intelligence	√		Winter
CS605 – Advanced Computer Architecture	√	√	---
CS606 – Computer Networks and the Internet	√	√	Winter
CS607 – Visual Computing	√	√	Winter
CS646 – Advanced Topics in Databases	√	√	---
CS651 – Mobile Computing and Data Management	√	√	Spring
CS653 – Computer Networks	√	√	Spring
CS655 – Advanced Parallel Processing II	√	√	---
CS656 – Computer Graphics - Modelling and Realism	√	√	Spring
CS657 – Wireless Networks	√		Spring
CS658 – Digital Video Processing	√		---
CS659 – Design with Embedded Processors	√	√	---
CS660 – Information Retrieval and Search Engines	√		Winter
CS662 – Machine Learning and Data Mining	√	√	---
CS663 – Computational Logic	√		---
CS664 – Systems Analysis and Verification	√		Spring
CS665 – Constraint Solving Methods	√		---
CS667 / MAI647 – Computational Neuroscience	√		Spring
CS668 – Mechanical Vision	√	√	Spring
CS673 – Algorithmic Game Theory	√		---
CS674 – System and Network Security	√	√	---
CS675 – Web Services and Service Oriented Computing	√	√	---
CS678 – Temporal Information Systems in Medicine	√	√	---
CS679 – Electronic Health	√	√	---
CS680 – Cognitive programming	√		Spring
CS681 – Advanced Software Reuse and Mining Software Repositories	√	√	---
CS682 – Advanced Security Topics	√	√	---
CS683 – Technology Entrepreneurship	√	√	---
CS699 – Special Topics in Computer Science	√		Winter

[1] No specialization courses exist. All courses are eligible.

Short Course Description

For the current academic year 2022-2023, each course description includes the name of the instructor, however, for the restricted courses that are not offered for the current academic year, 2022-2023, the name of the instructor who suggested the course is included. The language of instruction for all courses is Greek, unless English is specified as the instruction language.

CS601 Distributed Systems
Type: Restricted Choice for MCS, PM
Level: Graduate
Semester: Fall
Credit: 8 ECTS units
Instructor: Chr. Georgiou
Objectives: Familiarization with fundamental concepts and principles of distributed systems in both breadth and depth. Development of capabilities of designing, analyzing and programming distributed systems.
Content: Basic concepts and principles of distributed systems. Communication, processes and synchronization. Naming. Distributed file systems and distributed operating systems. Security and cryptography in distributed systems. Distributed shared memory and its consistency. Fault-tolerance. Distributed algorithms and distributed programming. Design and development of applications in distributed environments. Case-studies of specific distributed systems (eg. PlanetLab). Practical exposition with programming project or programming exercises.
Prerequisites: ----------
Bibliography:

1. A. S. Tanenbaum, M. van Steen, *Κατανεμημένα Συστήματα: Αρχές και Υποδείγματα*, Εκδόσεις Κλειδάριθμος, 2005.
   (English Version: A. S. Tanenbaum, M. van Steen, *Distributed Systems: Principles and Paradigms*, Pearson Education, 2nd Edition, 2007).
2. G. Goulouris, J. Dollimore, T. Kindberg, *Distributed Systems - Concepts and Design*, 4th Edition, Addison-Wesley, 2005.

Teaching methods: Lectures (3 hours weekly) and Recitation/Laboratory Lecture (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework (programming assignments).

CS603 Using Software Architectures for the Design and Development of Software Systems
Type: Specialisation Course for PM and Restricted Choice for MCS
Level: Graduate
Semester: Fall
Credit: 8 ECTS units
Instructor: …….
Objectives: The course will cover the basic principles, concepts and contemporary techniaues in using Software Architectures as the driving paradigm for the design and development of modern software systems. The role of Software Architectures in the broader field of Software Engineering will be addressed, with emphasis on the important paradigm of software reuse.
Content: Basic concepts. Design of a software architecture. Connectors. Modelling. Visual Representation. Design patterns for software architectures. Analysis and implementation. Non functional properties. Security and trust. Standards. The human factor. Domain-specific Software Engineering.
Prerequisites: Undergraduate course equivalent to the CS343 (Software Engineering)
Bibliography:

1. R. Taylor, N. Medvidovic, E. Dashofy, *Software Architecture: Foundations*, Theory, and Practice, 2010, Wiley

2. L. Bass, P. Clemens, R. Kazman, *Software Architecture in Practice*, 4th Edition, 2022, Pearson.

Teaching methods: Lectures (3 hours weekly) and Laboratory sessions (1.5 hours weekly).

Assessment: Midterm exam, final exam, homework and group project.

CS604 Artificial Intelligence

Type: Restricted Choice for MCS, PM
Level: Graduate
Semester: Fall
Credit: 8 ECTS units
Instructor: Y. Dimopoulos and Chr. Christodoulou
Objectives: This course covers specialized topics in Artificial Intelligence, such as modeling and solving constraint satisfaction and knowledge representation problems, symbolic learning, learning with various forms of neural networks, including deep learning and reinforcement learning.
Content: Introduction to Artificial Intelligence. Topics in Constraint Satisfaction. Satisfiability and Optimization in Logic. Answer Set Programming. Topics in Machine Learning, Data Mining, and Reasoning under Uncertainty. Introduction to Artificial Neural Networks. Single layer and Multi layer Perceptrons. Backpropagation learning algorithm. Deep Learning and Convolutional Neural Networks. Recurrent Neural Networks. Self-organizing Maps. Radial-basis Function Networks. Reinforcement Learning. Hopfield Neural Networks and Boltzmann Machines.
Prerequisites: ----------
Bibliography:

1. S. Russel, P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd Edition, Prentice Hall, 2014.
2. S. Haykin, *Neural Networks and Learning Machines*, 3rd Edition, Pearson Education, 2009.

Teaching methods: Lectures (3 hours weekly) and Recitation (1.5 hours weekly).

Assessment: Final exam, midterm exam and homework.

CS605 Advanced Computer Architecture

Type: Specialisation Course for PM and Restricted Choice for MCS
Level: Graduate
Semester: Fall
Credit: 8 ECTS units
Instructor: Y. Sazeidis
Objectives: Students will learn: (a) current methodology for performance evaluation and comparison of computer systems; (b) basic and advanced concepts in the organization of modern microprocessors; and (c) current trends in the computer architecture area. Also, with the use of different tools, presented in the course, students will perform research projects in certain computer architecture topics.
Content: Performance evaluation and comparison, as well as benchmarking programs; Basic microarchitecture concepts of modern processors; Pipelining, instruction-level parallelism, prediction, speculation, memory hierarchy, and static/dynamic instruction scheduling; Examples of modern processors; Current research projects in the area of computer architecture.
Prerequisites: Undergraduate course equivalent to the CS221 (Computer Organization and Assembly Programming) and undergraduate course equivalent to the CS222 (Operating Systems).
Bibliography:

1. J. Henessy, D. Patterson, *Computer Architecture: A Quantitative Approach*, 3rd Edition, Morgan Keufmann, 2002.
2. Selected research articles from the international literature.

Teaching methods: Lectures (3 hours weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam, homework (final project, intermediate assignments), class participation and brief presentation.

CS606 Computer Networks and the Internet
Type: Specialisation Course for PM and Restricted Choice for MCS
Level: Graduate
Semester: Fall
Credit: 8 ECTS units
Instructor: V. Vassiliou
Language: English
Objectives: Understanding (at a graduate level) of the basic concepts and matters regarding Computer Networks and the Internet. Familiarization with modern views of Computer Networks and exposure to the related open research problems.
Content: Introduction to Internet and Networking Technologies. TCP/IP suite of protocols, Quality of Service (QoS), New Networking Architectures. Protocols and Standards (e.g. DiffServ, IPv6, MPLS). Network Performance Evaluation (e.g. queueing theory, and simulation tools). Traffic Modeling and Traffic Engineering. Congestion Control and Resource Allocation. Network Design and Optimization.
Prerequisites: Undergraduate course equivalent to the CS324 (Communications and Networks)
Bibliography:

1. L. Peterson, B. Davies, *Computer Networks: A Systems Approach*, 5th Edition, Morgan Keufmenn, 2011.
2. J. F. Kurose, K. W. Ross, *Computers Networking – A Top Down Approach to the Internet*, 7th Edition, Addison-Wesley, 2016.

Teaching methods: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam and homework (Individual or Group Project and laboratory exercises).

CS607 Visual Computing
Type: Specialisation Course for PM and Restricted Choice for MCS
Level: Graduate
Semester: Fall
Credit: 8 ECTS units
Instructor: C. Pattichis / Y. Chrysanthou
Objectives: ----------
Content: Binary image processing, intensity transformations, the discrete Fourier transform, linear and nonlinear filtering, image compression, image analysis, basic principles of video processing. Basic principles of 3Dgraphics: polygonal representations, transformations, local and world
coordinate system, scene graph, camera and field of view specification, orthographic and perspective projection, clipping in 2D & 3D, polygon rasterization, back face elimination, visible surface determination with the Z-byffer method and Binary Space Partitioning Trees, local illumination - flat, Phong & Gouraud shading, real-time graphics, applications.
Prerequisites: Programming in C, basic linear algebra
Bibliography:

1. A. Watt, F. Policarpo, *The Computer Image*, Addison–Wesley, 1998.
2. R. C. Gonzalez, R. E. Woods, *Digital Image Processing*, 2nd Edition, Addison–Wesley, 2002.
3. M.Slater, A. Steed, Y. Chrysanthou, *Computer Graphics and Virtual Environments: From Realism to Real-Time*, Addison-Wesley, 2001.

Teaching methods: Lectures (3 hours weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam 40%, Midterm Exam 20%, Exercises and Project 40%.

CS646 Advanced Topics in Databases

Type: Specialisation Course for PM and Restricted Choice for MCS
Level: Graduate
Semester: Fall
Credit: 8 ECTS units
Instructor: D. Zeinalipour
Objectives: The main objectives of this graduate-level course are to provide an in-depth understanding of advanced concepts and research directions in the field of databases. The course is organized in three parts: (i) Fundamentals of Database Systems Implementation; (ii) Distributed, Web and Cloud Databases; (iii) Spatio-temporal Data Management, Sensor Data Management, other selected and advanced topics from the recent scientific literature.
Content: (i) Fundamentals of modern Database Management Systems (DBMSs): storage, indexing, query optimization, transaction processing, concurrency and recovery. (ii) Fundamentals of Distributed DBMSs, Web Databases and Cloud Databases (NoSQL / NewSQL): Semi-structured data management (XML/JSON, XPath and XQuery), Document data-stores (i.e., CouchDB, MongoDB, RavenDB), Key-Value data-stores (e.g., BerkeleyDB, MemCached), Introduction to Cloud Computing (GFS, NFS, Hadoop HDFS, Replication/Consistency Principles), "Big-data" analytics (MapReduce, Apache's Hadoop, PIG), Column-stores (e.g., Google's BigTable, Apache's HBase, Apache's Cassandra), Graph databases (e.g., Twitter's FlockDB) and Overview of NewSQL (Google's Spanner and Google's F1). (iii) Spatio-temporal data management (trajectories, privacy, analytics) and index structures (e.g., R-Trees, Grid Files) as well as other selected and advanced topics, including: Embedded Databases (sqlite), Sensor / Smartphone / Crowd data management, Energy-aware data management, Flash storage, Stream Data Management, etc.
Prerequisites: Undergraduate course equivalent to the CS342 (Database Systems)
Bibliography:

1. S. Abiteboul, I. Manolescu, P. Rigaux, M.-C. Rousset, P. Senellart, *Web Data Management*, Cambridge University Press, 2011.
2. R. Elmasri, S. Navathe, *Fundamentals of Database Systems*, 6th Edition, Addison-Wesley, 2011.
3. T. Özsu, P. Valduriez, *Principles of Distributed Database Systems*, 3rd Edition, Springer Press, 2011.
4. R. Ramakrishnan, J. Gehrke, *Database Management Systems*, 3rd Edition, McGraw-Hill, 2003.

Teaching methods: Lectures (3 hours weekly), and Laboratory sessions (1.5 hours weekly).

Assessment: Midterm, final exam and homework (assignments and presentation).

CS651 Data Management for Mobile Computing
Type: Specialization Course for PM and Restricted Choice for MCS
Level: Graduate
Semester: Spring
Credit: 8 ECTS units
Instructor: ----------
Objectives: Introduction to fundamental concepts, applications and limitations of mobile computing. Familiarization with practical applications and research topics of current interest in the field of Mobile Computing.
Content: Introduction (wireless technologies, architectures, applications, limitations). Software architectures for mobile computing. Theoretical models for mobile computing. Support for information recovery. Information Management. Dynamic redirection of computations. Indicative applications and open problems.
Prerequisites: Undergraduate course equivalent to the CS446 (Advanced Topics in Databases) and undergraduate course equivalent to the CS324 (Communications and Networks)
Bibliography:

1. E. Pitoura, G. Samaras, *Data Management for Mobile Computing*, Kluwer Academic Publisher, 1998.

Teaching methods: Lectures (3 hours weekly) and Recitation (1 hour weekly).
Assessment: Final exam, midterm exam and homework.

CS653 Computer Games Software Technology
Type: Specialisation Course for PM and Restricted Choice for MCS
Level: Graduate
Semester: Spring
Credit: 8 ECTS units
Instructor: ----------
Objectives: Study in depth of the technologies of Electronic Commerce. Introduction to the software technology of client/server systems of e-Commerce and to Business Models of e-Commerce.
Content: Game structure and design, computer animation, movement and deformation, interactive cameras, visual simulation of physically-based models, special effects using particle systems, collision detection, articulated characters, navigation and other behavioural models for autonomous characters.
Prerequisites: ---------
Bibliography:

1. R. Parent, *Computer Animation: Algorithms and Techniques*, Morgan Kaufmann, 2002.
2. A. Watt, M. Watt, *Advanced Animation and Rendering Techniques*, Addison-Wesley, 1992.
3. I. Millngton, *Artificial Intelligence for Games*, Morgan Kaufmann, 2006.

Teaching methods: Lectures (3 hours weekly) and Laboratory (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework.

CS655 Advanced Parallel Processing II
Type: Specialisation Course for PM and Restricted Choice for MCS
Level: Graduate
Semester: Spring
Credit: 8 ECTS units
Instructor: P. Trancoso
Objectives: Introduction to fundamental architectural concepts and methods for parallel program execution. Examine the challenges and difficulties for parallel computing.
Content: Support for parallel program execution, parallel architectures, different types of multiprocessor interconnection networks, compilation of parallel programs, and performance analysis of various parallel applications.
Prerequisites: Undergraduate course equivalent to the CS420 (Computer Architecture) or undergraduate course equivalent to the CS605 (Advanced Computer Architecture I) or the consent of the instructor.
Bibliography:

1. J. Henessy, D. Patterson, *Computer Architecture: A Quantitative Approach*. Third Edition, Morgan Kaufmann, 2002.
2. D. E. Culler, J. P. Singh. *Parallel Computer Architecture: A Hardware/Software Approach*, Morgan Kaufmann, 1998.
3. Selected research articles from the international literature.

Teaching methods: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework (group project and exercises).

CS656 Computer Graphics: Modeling and Realism
Type: Specialisation Course for PM and Restricted Choice for MCS
Level: Graduate
Semester: Spring

Credit: 8 ECTS units
Instructor: Y. Chrysanthou
Objectives: This course goes beyond the basics of digital image synthesis, looking at issues such as photo-realistic rendering, modeling and animation. A big component for this are the creation of realistic and detailed models as well as the faithful simulation of light transport. We will see how these can be applied to virtual and augmented reality. Students will acquire both the theoretical foundations as well as practical skills since a significant part of the course is the student project.
Content: Modeling, parametric and implicit surfaces, camera specification, projections of primitives. Graphics Pipeline. Local and global illumination, shadows, ray tracing and radiosity. Real-time rendering of large environments. Acceleration techniques.
Prerequisites: C Programming and basic Linear Algebra.
Bibliography:

1. M. Slater, A. Steed, Y. Chrysanthou, *Computer Graphics and Virtual Environments: From Realism to Real-Time*, Addison-Wesley, 2001.
2. A. Watt, *3D Computer Graphics*, 3rd Edition, Addison-Wesley, 2001.

Teaching methods: Lectures (3 hours weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework (group project and exercises).

CS657 Wireless Networks
Type: Restricted Choice for MCS, PM
Level: Graduate
Semester: Spring
Credit: 8 ECTS units
Instructor: A. Pitsillides
Objectives: Introduction to wireless networks (mobile/local/cellular/Ad-hoc/Sensor/IoTs) with an emphasis on the fundamental concepts and principles of the technologies which are important for the design, application, evaluation and development of these systems. The course will also cover new architectures and topologies, existing and proposed standards, as well as open research issues.
Content: Wireless environment, Interference and other problems in wireless communications, basic principles of wireless local and metropolitan area networks, and cellular wireless networks. Newer architectures and technologies of wireless networks and wireless communication (e.g., ad-hoc and sensor networks, VANETS). Resource management techniques, Next Generation wireless networks of 3rd, 4th and 5th generation (LTE, 4G, 5G, 6G), design and planning of wireless networks, protocols for wireless and mobile networks. Internet of Things (IoTs), new trends in Wireless Communication, such as Programmable Wireless Environments and Intelligent Surfaces.
Prerequisites: Undergraduate course equivalent to the CS324 (Communications and Networks)
Bibliography:

1. H Karl, A. Willing, *Protocols and Architectures for Wireless Sensor Networks*, Wiley, 2005.
2. K. Sohraby, D. Minoli, T. Znati, *Wireless Sensor Networks: Technology, Protocols, and Applications*, Published Online, 2006.
   K. Sohraby, D. Minoli, Taieb, *Wireless Sensor Networks: Technology, Protocols, and Applications*, 2006,

Teaching methods: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam and homework (including Individual or Group Project and laboratory exercises).

CS658 Digital Video Processing
Type: Restricted Choice for MCS, PM
Level: Graduate
Semester: ----------
Credit: 8 ECTS units
Instructor: C. Pattichis

Objectives: Basic familiarization with the application of current technology for processing and coding video.
Content: Basic of analog and digital video. Frequency domain analysis of video signals, spatial and temporal frequency response of the human visual system. Scene, camera, and motion modeling, 3D motion and projected 2D motion, models for typical camera/object motions. 2D motion estimation. Basic compression techniques. Waveform-based coding. Video compression standards (H.261 and H.263, MPEG-1, MPEG-2, MPEG-4, MPEG-7, MPEG-21).
Prerequisites: ----------
Bibliography:

1. Y. Wang, J. Ostermann, Y. Q. Zhang, *Video Processing and Communications*, Prentice Hall, 2002.

Teaching methods: Lectures (3 hours weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework (laboratory exercises, additional exercises, final study).

CS659 Design on Embedded Systems
Type: Specialisation Course for PM and Restricted Choice for MCS
Level: Graduate
Semester: ----------
Credit: 8 ECTS units
Instructor: ----------
Objectives: ---------
Content: A review of embedded system processors. Organization of embedded systems: CPUs, RAM, ROM, buses, peripherals, sensors, actuators, interfacing. Examples of widely used processors buses and peripherals. Interfacing with peripherals: sampling, interrupts, advantages and disadvantages. Process distribution between hardware and software. Tools for the development of embedded systems
and real-time operating systems. Hands-on experience with the development and implementation of embedded systems.
Prerequisites: Knowledge on the subjects of Digital Systems, Computer Organization and Assembly Programming
Bibliography:

1. F. Vahid, T. Givargis, *Embedded System Design: A Unified Hardware/Software Introduction*, John Wiley & Sons, 2002.
2. W. Wolf, *High-Performance Embedded Computing: Architectures, Applications and Methodologies*, Morgan Kaufman.
3. W. Wolf, *Computers as Components: Principles of Embedded Computing System Design*, Morgan Kaufman.
4. P. Raghavan, A. Lad, S. Neelakandan, *Embedded Linux System Design and Development*, Auerbach Publications

Teaching methods: Lectures (3 hours weekly) and Recitation (1 hour weekly) Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework.

CS660 Information Retrieval and Search Engines
Type: Restricted Choice for MCS, PM
Level: Graduate
Semester: Spring
Credit: 8 ECTS units
Instructor: G. Pallis
Objectives: The objective of this course is to examine the main computer science principles that lie behind Google and other search engines. To this end, the course will focus on basic and advanced techniques for text-based information systems: efficient text indexing; Boolean and vector space retrieval models; evaluation and interface issues; text classification and clustering. The course will also focus on Web search including crawling, link-based algorithms, and Web metadata.
Content: Introduction to Information Retrieval. Boolean Retrieval.Text encoding:

tokenisation, stemming, lemmatisation, stop words, phrases. Dictionaries and Tolerant retrieval. Index Construction and Compression. Scoring and Term Weighting. Vector Space Retrieval. Evaluation in information retrieval. Relevance feedback/query expansion. Text classification and Naive Bayes. Vector Space Classification. Flat andHierarchical Clustering.Web Search Basics.Web crawling and indexes. Link Analysis.
Prerequisites: Algorithms, Data Structures, Internet Technologies and Linear Algebra
Bibliography:

1. C. D. Manning, P. Raghavan, H. Schutze, *An Introduction to Information Retrieval*, Cambridge University Press, 2008.

Teaching methods: Lectures (3 hours weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework.

CS662 Machine Learning and Data Mining
Type: Specialisation Course for PM and Restricted Choice for MCS
Level: Graduate
Semester: --------
Credit: 8 ECTS units
Instructor: C. Pattichis
Objectives: Introduction of the fundamental principles, algorithms and techniques that support the development and implementation of data mining systems leading in the extraction of knowledge.
Content: Data Warehouse and OLAP Technology for Data Mining. Data Processing. Data Mining Primitives, Languages, and System Architectures. Concept Description: Characterization and Comparison. Mining Association Rules in Large Databases. Classification and Prediction. Cluster Analysis. Mining Complex Types of Data. Applications and Trends in Data Mining.
Prerequisites: --------
Bibliography:

1. J. Han, M. Kamber, *Data Mining – Concepts and Techniques*, Morgan Kaufmann, 2000.

Teaching methods: Lectures (3 hours weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework (case studies, exercises, oral presentation of a case study).

CS663 Computational Logic
Type: Restricted Choice for MCS, PM
Level: Graduate
Semester: ----------
Credit: 8 ECTS units
Instructor: A. Kakas
Objectives: Familiarization with fundamental concepts and applications of Computational Logic. Familiarization with current research trends in Computational Logic.
Content: Historical introduction. Review of Classical Logic. Abduction and induction. Knowledge representation and knowledge. Reasoning about Actions and Change. Application of Computational Logic. Declarative Programming. Autonomous Agents. Knowledge-based Robotics. Intelligent Information Integration.
Prerequisites: Undergraduate course equivalent to the CS324 (Communications and Networks)
Bibliography:

1. Selected research articles from the international literature.

Teaching methods: Lectures (3 hours weekly) and Recitation (1 hour weekly).
Assessment: Final exam and homework.

CS664 System Analysis and Verification
Type: Restricted Choice for MCS, PM
Level: Graduate
Semester: ----------
Credit: 8 ECTS units

Instructor: A. Philippou
Objectives: Familiarization with current technologies for modeling, analysis and verification of computer systems.
Content: Formal methods for system specification and analysis. Concurrent systems and interleaving and partial-order semantics. Transition systems and Kripke structures. Temporal logic (linear and branching). Automatic verification and model-checking. Process algebras: syntax, semantics, equivalence relations and axiom systems. Real-time system analysis (timed automata, timed process algebras and timed temporal logic). The tools SPIN and Concurrency Workbench.
Prerequisites: ----------
Bibliography:

1. D. Peled, *Software Reliability Methods*, Springer-Verlag, 2001.
2. C. Baier, J.-P. Katoen, Principles of Model Checking. MIT Press, 2008
3. L. Aceto, A. Ingólfsdóttír, K. G. Larsen, J. Srba, *Reactive Systems: Modelling, Specification and Verification*. Cambridge University Press, 2007.
4. Selected research articles from the international literature.

Teaching methods: Lectures (3 hours weekly) and Recitation (1 hour weekly).
Assessment: Final exam, midterm exam and homework.

CS665 Constraint Solving Methods
Type: Restricted Choice for MCS, PM
Level: Graduate
Semester: --------
Credit: 8 ECTS units
Instructor: Y. Dimopoulos
Objectives: A significant number of problems in Computer Science over a wide spectrum ranging from Computer Vision and Artificial Intelligence to the Management of Computer Networks and Scheduling are special cases of Constraint Satisfaction problems. This course covers advanced techniques for solving such problems and describes how they can be used in tackling real-life applications.
Content: Review of basic concepts from Constraint Satisfaction over Finite Domains. Advanced Consistency Techniques in Binary and non-Binary problems. Constraint Satisfaction and Propositional Logic: new algorithms and reduction techniques. Logic Programs with negations and the systems SMODELS and DLV. Relation between Constraint Satisfaction and Propositional Satisfiability. Redundant Constraints. Planning and Constraint Satisfaction. Satisfaction of Temporal Constraints. Introduction to Distributed Constraint Satisfaction. Problem solving with CHIP.
Prerequisites: CS604 or CS433
Bibliography:

1. R. Dechter, *Constraint Processing*, Morgan Kaufmann, 2003.
2. Selected research articles from the international literature.

Teaching methods: Lectures (3 hours weekly) and Recitation/Laboratory sessions (1 hour weekly).
Assessment: Final exam, midterm exam and programming exercises.

CS667 / MAI647 - Computational Neuroscience
Type: Restricted Choice for MCS, PM, MAI
Level: Graduate
Semester: Spring
Credit: 8 ECTS units
Instructor: Chr. Christodoulou
Objectives: Computational Neuroscience is an emerging and dynamically developing field aiming to elucidate the principles of information processing by the nervous system. This course aims to develop and apply computational methods for studying brain and behaviour as well as understanding the dynamics of the conscious mind..
Content: Introduction to Computational Neuroscience. Basic neurobiology: from the brain to single neurons; biophysics of single neurons. Conductance-based neuron models: the generation of action potentials and the Hodgkin and Huxley (HH) equations/model. Spiking

neuron models and response variability: leaky integrate-and-fire type neuron models, spike time variability. Two dimensional (2D) neuron models: reduction of the four dimensional HH model to a 2D model; phase plane analysis of 2D models/nullclines; FitzHugh-Nagumo model; neuronal dynamics. Modelling synapses/inputs to neurons. Neuron models beyond HH – more ion channels and their functions. Cable Theory: passive/active membranes; modelling axons and dendrites; action potential propagation. Compartmental models. Neural coding: firing rate; rate code; temporal code; neural operational modes – temporal integration/coincidence detection. Synaptic Plasticity: Hebbian learning; Spike-Timing Dependent Plasticity. Bottom-up/top-down modeling of the brain: modeling of self-control behaviour as an example of top-down modeling. Modelling consciousness.

Prerequisites: Linear Algebra, Differential Equations

Bibliography:

1. P. Dayan, L. Abbott, *Theoretical Neuroscience: Computational and Mathematical Modelling of Neural Systems*, MIT Press, 2001.
2. D. Sterratt, B. Graham, A. Gilles, D. Willshaw, *Principles of Computational Modelling in Neuroscience*, Cambridge University Press, 2011.
3. W. Gerstner, W. M. Kistler, R. Naud, L. Paninski, *Neuronal Dynamics: From single neurons to networks and models of cognition*, Cambridge: Cambridge University Press, 2014.
4. C. Koch, *Biophysics of Computation: Information Processing in Single Neurons*, Oxford University Press, 1998.
5. E. M. Izhikevich, *Dynamical Systems in Neuroscience: the Geometry of Excitability and Bursting*, MIT Press, 2007.

Teaching methods: Lectures (3 hours weekly) and Recitation/Laboratory sessions (1.5 hours weekly).

Assessment: Final exam, midterm exam and laboratory exercises /oral presentations of selected research papaers.

CS668 Mechanical Vision

Type: Specialisation Course for PM and Restricted Choice for MCS

Level: Graduate

Semester: Spring

Credit: 8 ECTS units

Instructor: C. Pattichis / Y. Chrysanthou

Objectives: The objective of this course is to understand the basic issues in mechanical vision and the major approaches that address them. Through the duration of the course, vision is treated as a process of inference from noisy and uncertain data in order to answer the question of how computers can understand the visual world of humans.

Content: Basic concepts and methodologies relating to the subject of Computer Vision. Image information, image processing, feature extraction. Image segmentation, clustering, multiple-image processing, case studies.

Prerequisites: ----------

Bibliography:

1. D. Forsyth, J. Ponce, *Computer Vision: A Modern Approach,* Prentice-Hall, 2003.
2. R. Hartley, A. Zeisserman, *Multiple View Geometry,* Cambridge University Press, 2003.
3. C. Bishop, *Pattern Recognition and Machine Learning*, Springer-Verlag, 2007.
4. O. Faugeras, Q.T. Luong, *Geometry of Multiple Images*, MIT Press, 2001.
5. B. Horn, *Robot Vision*, MIT Press, Cambridge, Massachusetts, 1986.

Teaching methods: Lectures (3 hours weekly) and Laboratory sessions (1 hour weekly).

Assessment: Final exam, midterm exam and homework (programming exercises).

CS670 Research Methodologies and Professional Practices in Computer Science

Type: Restricted Choice for MCS, PM

Level: Graduate

Semester: Spring

Credit: 4 ECTS units

Instructor: ……..

Objectives: Introduction to the methods and tools of Computer Science Research and technological culture. Familiarization with reading, reviewing and presenting of relevant literature. Technical writing. Literature surveying.
Content: Seminars/lectures in Computer Science and practice. Research or technical literature reviewing. Presentation of technical study.
Prerequisites: Successful completion of two graduate-level courses.
Bibliography:

1. Selected research articles from the international literature.
2. Course Presentation Slides (introductory and research).

Teaching methods: Lectures, research seminars and atomic assignments (summary of research seminars) and group study of a research subject under the supervision of a faculty member.
Assessment: Attendance and participation in lectures and a number of research seminars, written atomic studies, group study of a research subject and technical presentation of the group study. The course grade is Pass/Fail.

CS673 Algorithmic Game Theory
Type: Restricted Choice for MCS, PM
Level: Graduate
Semester: ----------
Credit: 8 ECTS units
Instructor: M. Mavronicolas
Objectives: ----------
Content Strategic games. Pure and mixed strategies, utilities, best responses, equilibrium concepts. Pure and mixed Nash equilibria, their refinements and generalisations. Classical existence theorems of equilibria and their algorithmic aspects. Algorithms and complexity of equilibrium searching. The complexity classes PLS and PPAD and their relation to equilibrium computation. Bimatrix games and
algorithms to compute their approximate equilibria. The Price of Anarchy and its variants. Analysis of the Price of Anarchy for both general and specific games (e.g., selfish routing games, congestion games, security games). Applications to realistic cases (e.g., social networks, Internet formation).
Prerequisites: Undergraduate course equivalent to the CS211 (Theory of Computation) and undergraduate course equivalent to the CS436 (Algorithms and Complexity)
Bibliography:

1. M. Mavronicolas, P. Spirakis, *Algorithmic Game Theory*, Springer, 2011, (book draft).
2. Selected research articles from the international literature

Teaching methods: Lectures (3 hours weekly) and Recitation/Laboratory sessions (1 hour weekly).
Assessment: Final exam, midterm exam and homework (theoretical assignments).

CS674 Networks and System Security
Type: Specialization Course for PM and Restricted Choice for MCS
Level: Graduate
Semester: ----------
Credit: 8 ECTS units
Instructor: A. Pitsillides, V. Vassiliou
Objectives: ----------
Content Introduction to security threats and attacks. Cryptographic and cryptanalysis techniques. Key exchange management (PKI). Network and Internet security protocols (IPSec, SSL/TLS). Identification and authentication standards (Kerberos, AAA). System security (Firewalls, IDS). Specific threats on end-systems (viruses, worms, trojan horses, stack overflow, rootkits). Identification of security vulnerabilities in software and operating systems. Checking of networks and applications for vulnerabilities, introduction to computer systems forensics. Security policies. Security management, ethical and legal issues in system security.
Prerequisites: Introductory graduate course equivalent to CS606 (Computer Networks and

the Internet)
Bibliography:

1. C. P. Pfleeger, S. L. Pfleeger, *Security in Computing*, Fourth Edition, Prentice Hall, 2006.
2. C. Kaufman, R. Perlman, and M. Speciner, *Network Security: PRIVATE communication in a PUBLIC world*, Second Edition, Prentice Hall, 2002.
3. M. Dowd, J. McDonald, J. Schuh, *The Art of Software Security Assessment*, Addison Wesley, 2006.

Teaching methods: Lectures (3 hours weekly) and Recitation/Laboratory sessions (1 hour weekly and 1.5 hours weekly, respectively).
Assessment: Final exam, midterm exam and homework (studies and/or laboratory assignments).

CS675 Web Services and Service Oriented Computing
Type: Specialisation Course for PM and Restricted Choice for MCS
Level: Graduate
Semester: ----------
Credit: 8 ECTS units
Instructor: ----------
Objectives: Understanding of the principles and the design of Web services, the programming of Web services and the protocols used. Understanding of the new architecture of microservices. Familiarization with knowledge representation via ontologies. Understanding of representation mechanisms.
Content: Introductory concepts. Relationship and difference between services and other related formalisms (distributed systems, component-based systems, etc.). Fundamental architectures and protocols (REST, SOAP, WSDL, UDDI). Fundamental development platforms (J2EE, JAX-RS, etc.). The microservices architecture. Problems and challenges. Information modeling and representation (ontologies, RDF and OWL protocols, etc.). Queries with SPARQL. LinkedData representation. Cooperative Information Systems and service composition. Business process modeling with WS-BPEL
Prerequisites: ----------
Bibliography:

1. M. P. Papazoglou, *Web Services: Principles and Technology*, 2nd edition, Pearson Prentice Hall, 2012.
2. R. Daigneau, Service Design Patterns Fundamental Design Solutions for SOAP/WSDL and RESTful Web Services, 2010.
3. D. Allemang, J. Hendler, Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL, Morgan Kaufman, 2011.

Teaching methods: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework (studies and/or laboratory assignments).

CS678 Temporal Information Systems in Medicine
Type: Restricted Choice for MCS, PM
Level: Postgraduate
Semester: Fall
Credit: 8 ECTS units
Instructor: E.Keravnou-Papailiou
Objectives: Often the question whether medicine is art or science is posed. The support of various medical tasks (diagnosis, prognosis, therapy planning and patient monitoring) through information systems, actively concerns a number of scientific communities (artificial intelligence, databases, biomedical engineering, medical informatics), where the temporal dimension constitutes an integral aspect. Recently the focus has changed, from applications based on knowledge, to applications based on data, and from systems that advise to systems that inform. The key challenge is no longer the mechanization of knowledge but the intelligent utilization of data. Medical problems and the deployment of medical data pose particular challenges, especially when the temporal dimension is a central component. The

course aims to present from a critical perspective the various aspects of Temporal Information Systems in Medicine (theoretical foundations, applications, open research topics) where such systems can be defined as information systems that can store, manage and query time-oriented clinical data and can support various reasoning tasks with respect to such data

Content: The significance of time in medicine. Modeling and reasoning with time (models of time and temporal entities). Requirements, ontologies and temporal reasoning models. General theories of time from the perspective of the medical domain. Temporal databases and their extensions for clinical data. Temporal abstraction of medical data (types of abstractions, time-oriented patient monitoring). Time and clinical diagnosis (diagnostic concepts, example applications, abductive reasoning using time-objects, temporal constraints). Automated support for clinical guidelines and protocols (time-oriented modeling of clinical guidelines). Research challenges.

Prerequisites: ----------

Bibliography

1. C. Combi, E. Keravnou-Papailiou, Y. Shahar, *Temporal Information Systems in Medicine*, Springer, 2010.
2. M. Fisher, D. Gabbay, L. Vila (eds), *Handbook of Temporal Reasoning in Artificial Intelligence*, Elsevier, 2005.
3. C. Bettini, S. Jajodia and S. X. Wang, *Time Granularities in Databases - Data Mining and Temporal Reasoning*, Springer, 2000.
4. I. Mani, J. Pustejovsky, R. Gaizauskas (eds.), *The Language of Time: A Reader*, Oxford University Press, 2005.
5. Articles from scientific journals, such as Artificial Intelligence in Medicine.

Teaching Methods: Lectures (3 hours weekly) and discussions/presentations in the form of seminars (1 hour weekly).

Assessment: Written examinations (midterm and final) and assignments (written essays and/or programming projects).

CS679 Electronic Health

Type: Specialisation Course for PM and Restricted Choice for MCS

Level: Graduate

Semester: Fall

Credit: 8 ECTS units

Instructor: ----------

Objectives: To introduce the student to the medical and clinical environment from the perspective of medical informatics and exploit the possibilities of using information technologies for modeling, prototyping, presenting and using the relevant data. To study and develop practical skills in building relevant intelligent information systems.

Content: Information retrieval from medical databases, data, medical records, live signals, and data mining using intelligent techniques. Study of application systems that are currently in use for managing medical data and suggest ways for better handling and building, medical knowledge bases, electronic health record, and decision support systems for the medical profession.

Prerequisites: --------

Bibliography:

1. J. H. van Bemmel, M. Musen (Edts), *Handbook of Medical Informatics*, Springer, 1997.
2. E.H. Shortliffe (Editor), L. M. Fagan, G. Wiederhold, L.E. Perreault, *Medical Informatics: Computer Applications in Health Care and Biomedicine*, Springer Verlag, Second, 2000.
3. L.Burke, B. Weill, *Information Technology for the health professionals*, Pentice Hall, 2000.

Teaching methods: Lectures (3 hours weekly) and Discussions/Presentations (1 hour weekly).

Assessment: Final exam, midterm exam and homework (studies and/or laboratory assignments).

CS680 Cognitive programming
Type: Restricted Choice for MCS, PM
Semester: Spring
Credit: 8 ECTS units
Instructor: A. Kakas
Objectives:
Content: Basic elements of cognitive science and the relation between logic and argumentation. Computation models for cognitive intelligence that follow representational models from cognitive psychology. The structure of knowledge and the human mechanism for common logic. The architecture of cognitive systems and their dynamic development cycle. Utilization of STAR, IBM Watson and other similar systems in the development of cognitive systems.
Prerequisites: ----------
Bibliography:
Teaching methods: ----------
Assessment: ----------

CS681 Advanced Software Reuse and Mining Software Repositories
Type: Specialisation Course for PM and Restricted Choice for MCS
Level: Graduate
Semester: Spring (or Fall)
Credit: 8 ECTS units
Instructor: G. Kapitsaki
Objectives: Understanding the usefulness of software reuse. Deepening in software design patterns. Undestanding the usefulness of data mining for software and being able to perform the process of collecting, preprocessing and processing data from software repositories and Q&A sites.
Content: Levels of reuse. Best practices for reuse. Software design patterns. Object-otiented patterns. Software repositories (e.g. GitHub). Search and retrieval. Data extraction and mining. Data mining steps. Data preprocessing and processing (e.g. in the R programming language). Use of dedicated APIs. Q&A sites, e.g. Stack Exchange, and data mining. Open source software. Open source licensing and legal issues. License compliance. Selecting licenses. Latest developments and research works.
Prerequisites: Basic understanding og object-oriented programming and software engineering process.
Bibliography:
1. T. Diamantopoulos, A. L. Symeonidis, *Mining Software Engineering Data for Software Reuse*, Springer, 2020.
2. M. Ezran, M. Morisio, C. Tully, *Practical Software Reuse*, Practitioner Series, 2002.
3. *Head First Design Patterns*, O'Reilly Media, 2004.
4. C. Horstmann, A *Practical Guide to Open Source Licensing*, Wiley, 2nd Edition, 2006.
5. Selected research papers and articles.

Teaching methods: Lectures (3 hours weekly), Recitation (1 hour weekly) and Laboratory sessions (1.5 hours weekly).
Assessment: Final and midterm exam, and homework (practical exercises and research assignments).

CS682 Advanced Security Topics
Type: Specialisation Course for PM and Restricted Choice for MCS
Level: Graduate
Semester: Spring
Credit: 8 ECTS units
Instructor: E. Athanasopoulos
Objectives: Crash-course introduction to applied cryptography, systems and network security and thoroughly covering of several advanced topics in applied cryptography, software vulnerabilities and memory errors, side channels, attacks and defenses, mobile

security, web security, network se-curity, privacy, and anonymity.
Content: Short introduction to security basics and then special topics are presented. Special topics include advanced cryptographic attacks in protocols, software exploi-tation through code-reuse (return-oriented programming, jump-oriented programming, and call-oriented programming), heap exploitation, side channels, advanced software hardening, exploit-ing special network protocols (DNS, NTP, etc.), complex attacks in network applications, and privacy issues. The course is seminar-based in part. Once the basics are introduced by the in-structor, students will study advanced papers in class and will have the opportunity to get a feel-ing of what are the important topics in modern security research.
Prerequisites: ----------
Bibliography:

1. A. J. Menezes, Π. C. van Oorschot, S. A. Vanstone. *Handbook of Applied Cryptography*, CRC Press.
2. R. Anderson. *Security Engineering: A Guide to Building Dependable Distributed Systems*, 2nd Edition, Wiley.
3. W. R. Cheswick, S. M. Bellovin, A. D Rubin. *Firewalls and Internet Security: Repelling the Wily Hacker*, 2nd Edition, Addison-Wesley Professional, 2003.
4. C. Anley, J. Heasman, F. Lindner, G. Richarte. *The Shellcoder's Handbook: Discovering and Exploiting Security Holes*, 2nd Edition. Wiley, 2007.
5. Selected research papers and articles.

Teaching methods: Lectures (3 hours weekly) and Recitation (1 hour weekly).
Assessment: Final exam, midterm exam, student paper presentation and homework (including laboratory as-signments).

CS683 Technology Entrepreneurship
Type: Restricted Choice for MCS, PM
Restricted Choice for Ph.D. candidates
Elective (Free) for students of other Departments
Level: Graduate
Semester: Fall
Credit: 8 ECTS units
Instructor: M. D. Dikaiakos
Language: English
Objectives: This course explores key challenges faced when one aspires to translate scientific and technological ideas into globally marketable products or services and embark on innovation-driven enterprise entrepreneurship, with a focus on software and Internet-based businesses. Students will develop knowledge and skills to understand and be able to address a variety of issues faced by startup founders and chief technology officers who need to innovate at the boundaries of information technology and business by understanding both perspectives.
Contents: Study and experimentation with acknowledged methodologies for innovation-driven entrepreneurship and associated tools, pursuing the translation of the students' ideas into entrepreneurial endeavors. In particular: the key stages of turning an idea or invention into a commercial product; the Lean Product Process and the Business Model Canvas methodologies in Information Technology and Scientific application contexts; the Disciplined Entrepreneurship methodology; techniques for the creative ideation and the design of software applications, products and services, such as Design Thinking, Innovators' Compass, and Sprint; fundraising and financing options for startups; the basics of incorporation and company structure; attracting talent, establishing and managing a team; tools for project and team management, collaboration, ideation, rapid prototyping: Trello, Slack, SimpleMind, Proto.io, Github, Google AdService, Google Cloud, Heroku, etc.; preparation of pitch decks, and pitch presentations in front of potential investors.
Prerequisites: ----------
Bibliography:

1. B. Aulet, *Disciplined Entrepreneurship*, Wiley, 2013.
2. B. Aulet, *Disciplined Entrepreneurship Workbook*, Wiley, 2017.

3. D. Olsen, *The Lean Product Playbook. How to Innovate with Minimum Viable Products and Rapid Customer Feedback*, Wiley, 2015.
4. A. Osterwalder et al, *Value Proposition Design: How to Create Products and Services Customers Want*, Wiley, 2014.
5. L. Klein, *UX for Lean Startups*, O'Reilly, 2013.
6. B. Horowitz, *The Hard Thing about Hard Things*, Harper Business, 2014.
7. St. G. Blank, *The Four Steps to the Epiphany. Successful Strategies for Products that Win*, Lulu, 2006.
8. Cl. Christensen, *The Innovator's Dilemma: When New Technologies Cause Great Firms to Fail (Management of Innovation and Change)*, Harvard Business Review Press, 2016.
9. E. Ries, *The Lean Startup*, Penguin, 2011.
10. T. Brown, *Change by Design*, Harper Collins, 2009.
11. T. Kelley, D. Kelley, *Creative Confidence*, Harper Collins, 2014.
12. P. Thiel, B. Masters, *Zero to One: Notes on Startups, or How to Build the Future*, Virgin Books, 2015.
13. J. Bezos, *The Everything Store: Jeff Bezos and the Age of Amazon*, Corgi, 2014.

Teaching methods: Lectures (3 hours weekly) and Recitation (1 hour weekly).
Assessment: Homework, semester group project (report and oral presentation).

CS699 Special Topics in Computer Science

Type: Restricted Choice for Ph.D. candidates (and conditionally for MCS, PM)
Level: Ph.D. and Advanced M.Sc.
Semester: .....
Credit: 8 ECTS units
Instructor: .....
Objectives: The objective of this course is to promote the research and technology culture in the graduate program of the Computer Science Department. The course focuses on a specific topic of Computer Science. Further information regarding the prerequisites, description, requirements and assessment of the course will be presented when the course is announced on the Department's website.
Content: The content of the course is according to the specific topic. Prerequisites: With consent of the Lecturer.
Prerequisites: The consent of the instructor.
Bibliography: ----------
Teaching methods: Lectures (3 hours weekly) and Recitation/Laboratory sessions (1.5 hours weekly).
Assessment: Final exam, midterm exam and homework (laboratory exercises and additional exercises).

MAI611 Artificial Intelligence Fundamentals
Type: Compulsory for MAI
Level: Graduate
Semester: Fall
Credit: 8 ECTS units
Instructor: E. Keravnou
Language: English
Objectives: The purpose of the course is to introduce students to the fundamental principles and techniques that underline software systems that exhibit "intelligent" behavior.
Content: Upon completion of this course, students will have acquired a good understanding of modern Artificial Intelligence, the problems that it addresses and the fundamental solution methods that it uses. More specifically the students will know the main knowledge representation techniques and reasoning methods that underlie artificial intelligence problem solving and be able to develop simple solvers for artificial intelligence systems.
Prerequisites: Knowledge of a high-level programing language, object-based data concepts and structures.
Bibliography:

1. S. Russel, P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th Edition, Pearson, 2021.

Teaching methods: Lectures, discussions of practical examples and (unsupervised) lab activities.
Assessment: Final exam, midterm exam and homework (theoretical and/or programming assignments).

MAI612 Machine Learning
Type: Compulsory for MAI
Level: Graduate
Semester: Fall
Credit: 8 ECTS units
Instructor: V. Vasiliades
Language: English
Objectives: Machine Learning (ML) is the branch of Artificial Intelligence (AI) that allows a computational system to improve itself through experience. It involves the development of systems that are trained to discover patterns in datasets, which can later be used to provide predictions on new data. ML is a rapidly evolving field, that has disrupted almost all scientific disciplines. This introductory course aims to provide a holistic view of ML covering sufficient breadth and depth, so that students understand the principles that drive most scientific and industrial AI innovations.
Content: After completing this course, the students will: (1) Understand how to structure ML projects and their lifecycle: from the data preparation phase to the development, evaluation and deployment phases; (2) Gain practical experience with various supervised learning models for regression, forecasting and classification problems; (3) Know how to implement unsupervised learning models for visualization, compression, clustering, anomaly detection and recommendation systems; (4) Understand what reinforcement learning is, how it can be used for sequential decision making problems and acquire hands-on experience with it.
Prerequisites: Basic programming skills, basic knowledge of linear algebra, calculus, probability and statistics.
Bibliography:

1. Bishop, C. M., *Pattern recognition and machine learning*, Springer, 2006.
2. Chollet, F., *Deep learning with Python*, Manning Publications, 2018.
3. Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y, *Deep learning*, MIT press, 2006.
4. Haykin, S., *Neural networks and learning machines*, 3rd Edition, Pearson Prentice Hall, 2009.
5. Howard, J., & Gugger, S., *Deep Learning for Coders with fastai and PyTorch*, O'Reilly Media, 2020.
6. Murphy, K. P., *Machine learning: a probabilistic perspective*, MIT press, 2012.

7. Powell, W. B., *Approximate Dynamic Programming: Solving the curses of dimensionality*, 2nd Edition, John Wiley & Sons, 2011.
8. Sutton, R. S., & Barto, A. G., *Reinforcement learning: An introduction*, 2nd Edition, MIT press, 2018.
9. Theodoridis, S., *Machine learning: a Bayesian and optimization perspective*, Academic press, 2015.

Teaching methods: The lectures will present the theory, while the laboratories will be complementary by providing the students with more content and tools that will help them understand the respective concepts both intuitively, as well as in more depth.
Assessment: Coursework and assignments (60%), final exam (40%).

MAI613 Research Methodologies and Professional Practices in AI
Type: Compulsory for MAI
Level: Graduate
Semester: Fall
Credit: 4 ECTS units
Instructor: K. Orphanou
Language: English
Objectives: The purpose of this course is to introduce students to the methods and tools of Artificial Intelligence Research, professional practices, and associated technological culture, bearing in mind EC's regulatory framework. Moreover, the course objectives encompass familiarization with reading, reviewing and presenting of relevant literature, technical writing and literature surveying.
Content: Upon completion of the course the students will be sufficiently conversant with the key methodological steps involved in carrying out research in AI and the safeguards for mitigating risks in potentially high-risk AI research and applications. Consequently, they will be familiar with the obligatory requirements for professional practices in AI to be characterized as secure, trustworthy and ethical. In addition, they will acquire experience in surveying some topic, writing a technical report on it and presenting it.
Prerequisites: Experience in orally presenting some topic to an audience and in writing a structured report (objectives, analysis, conclusions).
Bibliography:

1. Selected research articles from the international literature.
2. Course Presentation Slides.
3. Regulation of the European Parliament and of the European Council in Laying down harmonized rules for Artificial Intelligence ((COM (2021) 206 final).
4. P.R. Cohen, *Empirical Methodology for Artificial Intelligence*, MIT Press, 1995.

Teaching methods: Lectures, research/professional seminars, individual assignments (summary of research/professional seminars), and a group study of a research subject under the supervision of a faculty member.
Assessment: Attendance and participation in lectures and several research/professional seminars, written individual studies, group study of a research subject and technical presentation of the group study. The course grade is Pass/Fail.

MAI614 AI on the Edge Webinars I
Type: Compulsory for MAI
Level: Graduate
Semester: Fall
Credit: 2 ECTS units
Instructor: A. Kakas
Language: English
Objectives: The main objective of this course is for the students to be aware of the latest developments in AI and to appreciate the wide breath of study that AI requires.
Content: Students will become knowledgeable about a variety of the most recent developments in Artificial Intelligence and will have some appreciation of different AI centers world-wide.
Prerequisites: ----------
Bibliography: ----------

Teaching methods: A list of Webinars will be maintained and updated weekly from which each student can choose to attend. Students can also suggest other webinars outside the official list of the course, to be approved by the course instructor. The students submit a short report summarizing each talk they have attended and commenting on the significance of its topic to the development of AI. Interactive student discussions coordinated by industrial partners will also be included.
Assessment: The students are assessed by the quality of their reports for a minimum of 8 webinar talks they have attended.

MAI621 AI Ethics I
Type: Compulsory for MAI
Level: Graduate
Semester: Spring
Credit: 6 ECTS units
Instructor: Special Teaching Scientist
Language: English
Objectives: This course aims to raise awareness of the dangers that can arise from the development, deployment, and usage of intelligent autonomous systems and to introduce the students to socio-technical solutions for mitigating the risk of exhibiting unwanted non-ethical behaviour. Students will understand the basics of implementing systems that are not only high performing, but also adhere to our ethical socio-legal cultural values.
Content: The key learning outcomes of the course are: (1) Reflect upon the socio-ethical issues that arise upon the development, deployment, and usage of intelligent systems; (2) Critically discuss commonly occurring narratives and perspectives related to the use of AI; (3) Reason about the decisions made during a system's lifecycle and their relationship to accountability and responsibility; (4) Learn how to develop systems that exhibit a desired ethical behaviour and understand the main research challenges for this; (5) Understand how to judge and evaluate AI systems for their "ethicacy"; (6) Appreciate the socio-technical mechanisms for the governance of AI systems. By completing the above outcomes, the student will have a fundamental understanding of how intelligent systems influence—and are influenced by—our societies and of the socio-ethical responsibilities they have as developers and users of such tools.
Prerequisites: MAI611 Artificial Intelligence Fundamentals, basic technical knowledge of AI.
Bibliography:

1. V. Dignum, *Responsible Artificial Intelligence: How to Develop and Use AI in a Responsible Way*, Springer, 2019.
2. M. Coeckelbergh, *AI Ethics*, MIT Press, 2020.
3. D. Gunkel, *An Introduction to Communication and Artificial Intelligence*, Willie, 2020.
4. C. O'Neil, *Weapons of Math Destruction*, Crown Books, 2016.
5. F. Pasquale, *The Black Box Society: The Secret Algorithms That Control Money and Information*, Belknap Press, 2014.
6. Papers, as reading material, will be made available to students on a weekly basis.

Teaching methods: A variety of teacher-led and student-led activities. Weekly lectures will introduce and provide overview of topics. Students will conduct self-study of the weekly material. Students will be given the opportunity to participate in problem-based solving group exercises, where they will conduct critical analysis and debate timely issues related to AI ethics. In unsupervised technical labs, students will be given the opportunity to test technical solutions for compliance checking and implement machine ethics, i.e., agents with moral reasoning.
Assessment: Two major assignments (one group, one individual), final exam.

MAI622 AI Entrepreneurship
Type: Compulsory for MAI
Restricted Choice for MDS, MCS, PM
Level: Graduate
Semester: Spring

Credit: 8 ECTS units
Instructor: M. D. Dikaiakos
Language: English
Objectives: This course seeks to help students explore and master key concepts and challenges of relevance to AI and Data-driven entrepreneurship. The course introduces students to the world of AI entrepreneurship through case studies that demonstrate successes, failures and challenges. The course provides also an overview of and an introduction to key steps to develop a company, design a business model, explore product-market fit, manage intellectual property, and attract investment. Students will explore acknowledged innovation-driven entrepreneurship methodologies and experiment with them and associated tools to pursue the translation of their ideas into entrepreneurial endeavors. The course examines issues faced by Startup Founders and Chief Technology Officers who need to innovate at the boundaries of AI, Information Technology and Business by understanding all perspectives.
Content: The course will comprise weekly live and recorded lectures by the professor and by invited speakers on various aspects of entrepreneurship and innovation. The students will be required to establish teams and work on an idea, producing a business plan and a prototype of an MVP, and several writeups for class readings and invited lectures. Lectures will cover Case Studies in AI Entrepreneurship, Basic concepts in Entrepreneurship and Innovation, and elements of Preparatory Analysis for establishing a startup company, Setting up a company, Value Proposition, Market Analysis and Competition, Business Modeling for AI Products and Services, Customer acquisition and Sales.
Prerequisites:
Bibliography:

1. B. Aulet, *Disciplined Entrepreneurship*, Wiley, 2013.
2. B. Aulet, *Disciplined Entrepreneurship Workbook*, Wiley, 2017.
3. A. Osterwalder et al, *Value Proposition Design: How to Create Products and Services Customers Want*, Wiley, 2014.
4. A. Fontana, *The AI-First Company: How to Compete and Win with Artificial Intelligence*, Penguin, 2021.
5. P. Thiel, B. Masters, *Zero to One: Notes on Startups, or How to Build the Future*, Virgin Books, 2015.

Teaching methods: Lectures (3 hours weekly), recitation sessions (1 hour weekly), team projects (all semester).
Assessment: Group project report and presentation, writing assignments.

MAI601 AI Camp
Type: Restricted choice for MAI
Level: Graduate
Semester: Summer
Credit: 4 ECTS units
Instructor: ----------
Language: English
Objectives: Raising the awareness that students are studying in an EU wide program. Awareness of the career opportunities in the various countries and further studies at the partner universities. Initiating joint supervision and joint research.
Content: Learning to collaborate at a distance in a multi-national group. Learning to present and to defend his or her own work in an international community of peers and experts. Improved social skills of students.
Prerequisites:
Bibliography:

1. Students will be given reading material to help them with their project preparation and with their presentations at the summer camp.

Teaching methods: One-week retreat in some relatively isolated place so that students and faculty from the universities participating in the network of MAI4CAREU can have a close working interaction. Before the event the students who will participate are given projects to carry out in groups of 3-4 students from different universities. At the event they are asked to finalize their project and present it to the community attending the camp. Groups of

students will also jointly present research studies of some topic in AI. There will also be a special session where students can give feedback on the Masters program that they are attending. Faculty and SMEs will present to the students, current career and further research opportunities in the institutions and other links they have in the EU.
Assessment: Students will be assessed on their presentations at the event and on their related reports. This will be a Pass/Fail assessment.

MAI631 AI Ethics II and AI Policy Making
Type: Compulsory for MAI
Level: Graduate
Semester: Fall
Credit: 4 ECTS units
Instructor: ----------
Language: English
Objectives: The main purpose of this course is to help the students understand in depth the major ethical issues that concern various sectors of the society and how these are reflected into AI policy by governments and organizations, particularly by EU policy on AI. The students will understand how research into AI Ethics feeds into policy and how policy requirements affect the development of AI systems.
Content: The key learning outcomes of the course are: (1) Grasp the importance role of analysing and mitigating socio-ethical issues by forming AI policies at various levels in the society; (2) Acquire an in-depth knowledge about governmental and intergovernmental AI policy initiatives—particularly by and within the European Union; (3) Understand the relationship between ethical governance and regulatory frameworks; (4) Explore how AI Ethics affects the strategy of large and small companies for the development of their AI systems; (5) Develop a critical understanding of socio-technical mechanisms for the governance of AI systems, recognizing the drawbacks and benefits of each approach; (6) Reflect upon how research in AI feeds into AI policy and how AI policy creates new needs for research into AI Ethics. By completing these outcomes, the student will be able to understand and comment on AI policies and the research challenges emerging from them.
Prerequisites: MA621 AI Ethics I, basic technical knowledge of AI.
Bibliography:

1. EU AI Policy documents such as: The ethics of artificial intelligence: Issues and initiatives; Ethics guidelines for trustworthy AI; Artificial intelligence: From ethics to policy.
2. Other AI Policy resources: OECD AI Policy Observatory; AI Policy Forum at MIT.
3. Open Lecture Resources: Ethics in AI seminar series at Oxford; Online Speaker Series on the Ethics of Argumentation; AI Ethics Seminars at Chalmers
4. Journal of AI Ethics, Springer (launched in February 2021).

Teaching methods: Bi-weekly lectures will introduce and provide overview of topics. Students will conduct a self-study on a weekly basis by attending seminars with guest speakers on AI Ethics & Policy issues, studying policy documents, and applying AI policies to their AI projects (e.g., undergraduate dissertations, assignments from other projects, etc). Students will be required to present the results of their study in reports and/or presentations to the class.
Assessment: Bi-weekly short assignments, one group project and final essay exam.

MAI632 AI on the Edge Webinars II
Type: Compulsory for MAI
Level: Graduate
Semester: Fall
Credit: 2 ECTS units
Instructor: G. Pallis
Language: English
Objectives: The main objective of this course is for the students to be aware of the latest developments in AI and to appreciate the wide breath of study that AI requires.
Content: Students will become knowledgeable about a variety of the most recent

developments in Artificial Intelligence and will have some appreciation of different AI centers world-wide.
Prerequisites: ----------
Bibliography: ----------
Teaching methods: A list of Webinars will be maintained and updated weekly from which each student can choose to attend. Students can also suggest other webinars outside the official list of the course, to be approved by the course instructor. The students submit a short report summarizing each talk they have attended and commenting on the significance of its topic to the development of AI. Interactive student discussions coordinated by industrial partners will also be included.
Assessment: The students are assessed by the quality of their reports for a minimum of 8 webinar talks they have attended.

MAI641 Master Thesis
Type: Restricted choice for MAI
Level: Graduate
Semester: Fall
Credit: 16 ECTS units
Instructor: ----------
Language: English
Objectives: The main objective of this course is to enable the students to develop deeper knowledge, understanding, capabilities and attitudes in the context of the programme of study. The thesis should be written at the end of the programme and offers the opportunity to delve more deeply into and synthesize knowledge acquired. The thesis will place emphasis on the technical and/or scientific aspects of the subject matter.
Content: (1) Considerably more in-depth knowledge of the major subject/field of study, including deeper insight into current research and development work; (2) Deeper knowledge of methods in the major subject/field of study; (3) A capability to contribute to research and development work; (4) The capability to use a holistic view to identify, formulate and deal with complex issues critically, independently and creatively; (5) The capability to plan and use adequate methods to conduct qualified tasks in given frameworks and to evaluate this work; (6) The capability to create, analyze and critically evaluate different technical solutions; (7) The capability to integrate knowledge critically and systematically; (8) The capability to clearly present and discuss the conclusions as well as the knowledge and arguments that form the basis for these findings in written and spoken English; (9) A consciousness of the ethical aspects of research and development work.
Prerequisites: A student must complete successfully courses, totaling at least 45 ECTS credits from the MSc AI Programme.
Bibliography: The bibliography of this course will be determined by the Research Advisor.
Teaching methods: For a student to undertake a Master Thesis a Research Advisor, based on the rules of the University Senate, is assigned to the student before s/he submits the Thesis Proposal. The Thesis must deal with a research topic or a technical issue. It must be of some original contribution or show a thorough and clear understanding of some special topic. A student participating in the AI Camp and/or completing successfully an industrial internship may discuss the possibility of doing his/her Master Thesis in collaboration with an industrial partner.
Assessment: The Master Thesis is submitted at the Department and defended within the time period decided by the Departmental Council and in accordance with the relevant rules and regulations of the Department and the University.

MAI623 Natural Language Processing
Type: Restricted choice for MAI
Level: Graduate
Semester: Spring
Credit: 8 ECTS units
Instructor: Special Teaching Scientist
Language: English

Objectives: Natural language processing (NLP) seeks to provide computers with the ability to intelligently process human language, extracting meaning, information, and structure from text, speech, web pages, and social networks. The goal of this course is to provide the fundamental aspects of NLP systems, as well as introduce recent advancements in the field of NLP and Deep Learning. The course is organized into two parts: (1) Fundamental knowledge, concepts, and techniques of NLP, and (2) introduction to Deep Learning methods for NLP.
Content: The students who complete this course successfully will be able to: (1) Comprehend various fundamental concepts of NLP: Text processing (normalization, lemmatization, stemming, etc.), language models (N-Grams), word representation (word embeddings), and text classification with Machine Learning; (2) Familiarize with known NLP tasks: Named Entity Recognition (NER), Part-of-Speech tagging (PoS), Dependency and Syntax parsing; (3) Employ Machine Learning (ML) techniques for text classification (e.g., Naive Bayes) and be able to properly apply the NLP feature engineering process; (4) Extend their knowledge with advanced methods in NLP and Deep Learning: Word Vectors, Word2vec algorithm, BERT; (5) Apply their knowledge on real-world research applications of NLP and recognize the societal impact in cases of misinformation and hate-speech identification; (6) Design efficient and effective NLP solutions to a variety of problems, using state-of-the-art tools.
Prerequisites: MAI612 Machine Learning, familiarity with Python.
Bibliography:

1. Dan Jurafsky and James H. Martin, *Speech and Language Processing*
2. Jacob Eisenstein, *Natural Language Processing*
3. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*, 1st Edition

Teaching methods: Students will meet the expected learning outcomes through participation in lectures, active participation and in class discussions, and actual practice with programming assignments and the final project. The lectures will be hybrid, with the possibility of both physical and virtual presence of the students.
Assessment: Student progress is evaluated continuously through class participation and the assessment of at-home assignments, group project deliverables, and final exam. The final grade is based on the following formula (the percentages are indicative): Assignments and Participation: 30%; Final Project: 40%; Final Exam: 30%.

MAI642 Deep Learning
Type: Restricted choice for MAI
Level: Graduate
Semester: ----------
Credit: 8 ECTS units
Instructor: ----------
Language: English
Objectives: The objective of this course is to provide a concrete understanding of the fundamental concepts of deep learning used for computer vision applications for image and video processing and understanding. Deep Learning is a key driving force behind Artificial Intelligence (AI) breakthroughs over the past few years and a paradigm shift in most computer vision tasks performed today.
Content: A well-balanced understanding of deep learning theoretical concepts and computer vision applications materialized using contemporary deep learning architectures and software tools.
Prerequisites: MAI612 Machine Learning, Digital Image Processing Fundamentals, basic programming skills.
Bibliography:

1. I. Goodfellow, Y. Bengio, A. Courville, Y. Bengio, *Deep learning*, MIT Press, 2016. Freely Available: https://www.deeplearningbook.org/
2. Adrian Rosebrock, *Deep Learning for Computer Vision with Python*, 2017.
3. A. Geron, *Hands-On Machine Learning with Scikit-Learn & TensorFlow Concepts, Tools, and Techniques to Build Intelligent Systems*, 2nd Edition, O'Reilly Media, 2019.

Teaching methods: The course balances theory and practice as it involves a significant amount of example case studies that aim to complement both the lectures and hands-on lab training. The students are expected to apply the learning outcomes throughout the duration of the course during the assigned coursework.
Assessment: Assignments and Projects (60%); Mid-term exam (10%); Final exam (30%).

MAI643 AI in Medicine
Type: Restricted choice for MAI
Level: Graduate
Semester: Spring
Credit: 8 ECTS units
Instructor: E. Keravnou / K. Orphanou
Language: English
Objectives: The medical domain has presented key challenges to the AI community from the early days of AI research. It is not an exaggeration to say that this pioneering work, particularly in medical expert systems, and its undisputable successes, some in real-life settings, has helped both in restoring confidence in the promise of AI, that at some point was disturbed after its failure to deliver fully on the very ambitious initial goals that it had set, and in paving the way towards more viable paths harnessing the mechanization of knowledge and human expertise. The aim of this elective course is to familiarize students with the past, present and future of Artificial Intelligence in Medicine, illustrating the discussion with several case studies, and pinning down the human-centric and ethical aspects underlying the given applications.
Content: Upon completion of the course the students will have a good understanding, from a critical perspective, of the span of applications of AI methods and techniques in the medical domain, and the methodologies used in developing such applications. More specifically the students will understand the importance of time in medical information systems and how time can be modelled, be conversant with data-driven clinical decision-making, and grasp the regulatory, social, ethical and legal issues of Artificial Intelligence in Medicine.
Prerequisites: MAI611 AI Fundamentals.
Bibliography:

1. A.C. Chang, *Intelligence-Based Medicine: Artificial Intelligence and Human Cognition in Clinical Medicine and HealthCare*, Academic Press, 2020.
2. C. Combi, E. Keravnou-Papailiou, Y. Shahar, *Temporal Information Systems in Medicine*, Springer, 2010.
3. L. Xing, M.L. Giger, J.K. Min (editors), *Artificial Intelligence in Medicine: Technical Basis and Clinical Applications*, Academic Press, 2021.
4. Panesar, *Machine Learning and AI for Healthcare: Big Data for Improved Health Outcomes*, Apress, 2021.
5. T. Lawry, *AI in Health: A Leader's Guide to Winning the New Age of Intelligent Health Systems*, CRC Press, 2020.
6. N. Lavrac, E.T. Keravnou, B. Zupan (editors), *Intelligent Data Analysis in Medicine and Pharmacology*, Kluwer Academic Publishers, 1997.
7. E.T. Keravnou (editor), *Deep Models for Medical Knowledge Engineering*, Elsevier Science Publishers, 1992.
8. Scientific papers from thematic and/or standard issues of relevant journals, primarily the journal Artificial Intelligence in Medicine (AIME) published by Elsevier.

Teaching methods: Lectures and discussions particularly around the presented case studies. Students would be strongly guided to view all topics presented and discussed with a critical eye.
Assessment: Final exam, midterm exam and project assignments.

MAI644 Computer Vision
Type: Restricted choice for MAI
Level: Graduate
Semester: Fall

Credit: 8 ECTS units
Instructor: M. Averkiou
Language: English
Objectives: This course aims to build a fundamental understanding of classic computer vision, starting at extracting and describing features such as edges and corners from images, moving to mid-level tasks such as model fitting and image stitching, then, high-level tasks such as semantic segmentation, recognition, and detection, and ending with motion and extracting scene geometry from images.
Content: At the end of this course, students should: (1) Understand the fundamentals of classic computer vision; (2) Be able to identify the recent trends and developments in computer vision; (3) Identify limitations of the current state of the field and the immense potential for commercial applications of computer vision; (4) Apply mathematical methods in a rigorous manner in order to solve computer vision tasks; (5) Know how an image is formed and how cameras work; (6) Know what features are and how they are extracted from an image; (7) Know what edge and corner detection is; (8) Know how features are described, stored and how they are used to solve computer vision problems; (9) Understand classic computer vision algorithms such as RANSAC or Normalized cuts as well as methods such as PCA; (10) Be confident in camera models and projective transformations; (11) Know what camera extrinsic and intrinsic parameters are how to perform camera calibration; (12) Understand how stereo and multi-view reconstruction works, and be able to appreciate structure from motion algorithms; (13) Understand high-level tasks such as segmentation, recognition, detection, tracking.
Prerequisites: MAI612 Machine Learning, linear algebra, data structures and algorithms.
Bibliography:

1. David A. Forsyth, Jean Ponce, *Computer Vision A Modern Approach*, 2nd Edition, Prentice Hall, 2012.
2. Richard Szeliski. *Computer Vision: Algorithms and Applications*, 2nd Edition, Springer, 2020
3. Hartley, Andrew Zisserman, *Multiple View Geometry in Computer Vision*, Academic Press, 2002.
4. Christopher Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

Teaching methods: Lectures and labs.
Assessment: Final exam (40%); Mid-term exam (30%); Coursework and assignments (30%). To qualify one must: (i) Hand in all assignments and coursework; (ii) Achieve at least 50% weighted average in the mid-term and final exam; (iii) Achieve at least 50% overall.

MAI645 Machine Learning for Graphics and Computer Vision
Type: Restricted choice for MAI
Level: Graduate
Semester: Spring
Credit: 8 ECTS units
Instructor: A. Aristidou
Language: English
Objectives: This course will offer an introduction to machine learning algorithms, the use of deep learning and its applications in computer vision and graphics. The course will also operate as a graduate-level seminar with weekly readings (1 hour per week), summarizations, and discussions of recent papers.
Content: Participants will explore the latest developments in neural network research and deep learning models that are enabling highly accurate and intelligent computer vision and graphics systems. By the end, participants will: (1) Be familiar with fundamental concepts and applications in computer vision and graphics; (2) Grasp the principles of state-of-the art deep neural networks; (3) Gain knowledge of high-level vision tasks, such as object recognition, scene recognition, face detection and human motion categorization; (4) Gain knowledge of high-level graphics tasks, such as composite image generation, style transfer, motion reconstruction, and motion synthesis; (5) Develop practical skills necessary to build highly accurate, advanced computer vision and graphics applications.
Prerequisites: Experience in programming with Python; Experience with linear algebra,

calculus, statistics and probability.
Bibliography:

1. Ian Goodfellow, Yoshua Bengio, Aaron Courville, *Deep Learning*, MIT Press, 2016.
2. Steve Holden, *Computer Vision: Advanced Techniques and Applications*, CLANRYE INTERNATIONAL, 2019.
3. Christopher Bishop, *Pattern Recognition and Machine Learning*, Springer, 2016.

Teaching methods: 3 hours lectures; 1.5 hours lab; 1 hour for recitation and discussion on recent papers.
Assessment: Exams (30%); Student paper presentations (15%); Programming assignments (25%); Final course project (30%).

MAI646 Cognitive Programming for Human-Centric
Type: Restricted choice for MAI
Level: Graduate
Semester: Fall
Credit: 8 ECTS units
Instructor: A. Kakas
Language: English
Objectives: The goal of the course is to introduce students into the new framework for Cognitive Computing for the development of Cognitive Systems that serve the needs of Human-centric AI. The theoretical understanding of the challenges of such cognitive systems and the development of knowledge for their practical application will be discussed.
Content: The key learning outcomes of the course are for students to master: (1) Properties and Design of Cognitive Systems; (2) Automated Cognitive Decision Making; (3) Argumentation for Human Cognitive Reasoning; (4) Computational Argumentation; (5) Learning & Reasoning in Cognitive Systems; (6) Software Methodology for Cognitive Assistants.
Prerequisites: Basic knowledge of AI.
Bibliography:

1. David Vernon, *Artificial Cogntive Systems*, MIT Press, 2014.
2. Antonio Lieto, *Cognitive Designs for Artificial Minds*, Routledge, 2021.
3. Journal of Cognitive Systems Research, Elsevier.
4. Journal of Advances of Cognitive Systems.
5. Journal of Computational Cognitive Science.
6. *Principles of Synthetic Intelligence*, Oxford University Press, 2009.
7. Research Documents on Cognitive Computing.

Teaching methods: Weekly lectures will introduce and provide overview of the course topics. In addition, there will be a running project throughout the course for the students to develop a Cognitive Assistant of their own application choice. The students will also undertake a bibliography assignment to review a topic related to Cognitive Computing from other disciplines. Small exercises help the student develop knowledge representation skills in argumentation form and programming skills in the Gorgias system of argumentation.
Assessment: Two major projects, one on research study and one developing a Cognitive Assistant. Four small assignments and a final exam.

MAI647 Computational Neuroscience
Type: Restricted choice for MAI
Level: Graduate
Semester: Spring
Credit: 8 ECTS units
Instructor: ----------
Language: English
Objectives: Computational Neuroscience is an emerging and dynamically developing field aiming to elucidate the principles of information processing by the nervous system. This course aims to develop and apply computational methods for studying brain and behaviour

as well as understanding the dynamics of the conscious mind.
Content: The learning outcomes for the students are the following: (1) Understand and be able to explain the fundamental principles of information processing by neural systems; (2) Appreciate the importance of computational neuronal models in the quest of understanding the brain and the fact that many aspects of neuroscience cannot be understood without appropriate computational modeling framework; (3) Understand the most important biophysical neuronal models and the different levels of description and complexity in computational neuronal modelling from the level of the single neuron to that of neural networks; (4) Understand neuronal dynamics and learn how high dimensional neuronal models can be reduced to low dimensional neural models; (5) Understand how experimentally recorded physiological signals enable us to understand the functionality of neurons/systems in the brain and how statistical approaches help in the analysis of such data; (6) Be able to implement/simulate basic computational neuronal models through programming; (7) Become familiar and be able to use various computational neuroscience simulation software packages for modelling complex biophysical models and experimentally observed phenomena; (8) Be able to grasp the importance of high-level modelling abstraction from the underlying neuronal principles for understanding brain behaviours; (9) Critical reading and discussion of recently published scientific papers.
Prerequisites: Linear algebra, differential equations, programming.
Bibliography:

1. P. Dayan, L. Abbott, *Theoretical Neuroscience: Computational and Mathematical Modelling of Neural Systems*, MIT Press, 2001.
2. W. Gerstner, W. M. Kistler, R. Naud, L. Paninski, *Neuronal Dynamics: From single neurons to networks and models of cognition*, Cambridge University Press, 2014.
3. D. Sterratt, B. Graham, A. Gilles, D. Willshaw, *Principles of Computational Modelling in Neuroscience*, Cambridge University Press, 2011.
4. W. Gerstner, W. M Kistler, *Spiking Neuron Models: Single Neurons, Populations and Plasticity*, Cambridge University Press, 2002.
5. C. Koch, *Biophysics of Computation: Information Processing in Single Neurons*, Oxford University Press, 1998.
6. T. Trappenberg, *Fundamentals of Computational Neuroscience*, 2nd Edition, Oxford University Press, 2010.

Teaching methods: Lectures (3 hours weekly), recitation (1 hour weekly) and laboratory (2 hours weekly).
Assessment: Final exam, midterm exam and laboratory exercises/oral presentations of selected research papers.

MAI648 Human-Centered Intelligent User Interfaces
Type: Restricted choice for MAI
Level: Graduate
Semester: Fall
Credit: 8 ECTS units
Instructor: M. Belk
Language: English
Objectives: The purpose of the course is to introduce students to fundamental principles and methods within the intersection of Artificial Intelligence and Human-Computer Interaction aiming to design and develop more efficient and effective user interfaces through the use of intelligent computation methods.
Content: Upon completion of this course, students will have acquired: (1) An in-depth understanding of theoretical and practical aspects of intelligent user interfaces; (2) Skills to design, develop and evaluate intelligent interactive systems by considering a variety of human factors, such as human cognitive and emotional characteristics for improving the efficiency, effectiveness and user experience in interactive systems; (3) Abilities to synthesize and evaluate the potential of this knowledge in relation to deploying intelligent user interfaces in real-life applications.
Prerequisites: Basic knowledge on AI and HCI, general programming knowledge.
Bibliography:

1. Germanakos, P., Belk, M, *Human-Centered Web Adaptation and Personalization - From Theory to Practice*, Human-Computer Interaction Series, Springer, 2016.
2. Brusilovski, P., Kobsa, A., Nejdl, W, *The Adaptive Web: Methods and Strategies of Web Personalization*, Springer, 2007.
3. Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., Diakopoulos, *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, 6th Edition, Pearson, 2017.
4. Preece, J., Sharp, H., Rogers, Y., *Interaction Design: Beyond Human-Computer Interaction*, 4th Edition, Wiley, 2015.

Teaching methods: Lectures covering the theoretical foundations of intelligent user interfaces, discussion of practical examples, and lab activities for designing and implementing intelligent user interfaces.
Assessment: Final exam, midterm exam and homework (theoretical and programming assignments).

MAI649 Principles of Ontological Databases
Type: Restricted Choice for MAI
Level: Graduate
Semester: Spring
Credit: 8 ECTS units
Instructor: A. Pieris
Language: English
Objectives: Nowadays we need to deal with data that is very large, heterogeneous, distributed in different sources, and incomplete. At the same time, we have very large amounts of knowledge about the application domain of the data in the form of ontologies that can be used to provide end users with flexible and integrated access to data. This gave rise to ontological databases, which lie at the intersection of traditional databases, and knowledge representation and reasoning. The purpose of the course is to introduce students to the principles of ontological databases and demonstrate the importance of studying data-intensive problems in a mathematically rigorous way, as well as the implications of such studies for real-life applications.
Content: Towards the main goal of the course, it is vital to first cover the principles of relational databases, without taking ontologies into account, on top of which the principles of ontological databases are built. In particular, the course will cover the following topics: (1) Relational model: data model, relational algebra, relational calculus (first-order queries), first-order query evaluation, static analysis of first-order queries; (2) Conjunctive queries (CQs): syntax and semantics, CQ evaluation, static analysis of CQs, minimization of CQs, acyclicity of CQs, evaluation of acyclic CQs (Yannakaki's algorithm), semantically acyclic CQs and their evaluation; (3) Adding recursion - Datalog: inexpressibility of recursive queries, syntax and semantics of Datalog, Datalog query evaluation, static analysis of Datalog queries; (4) Ontological databases: rule-based ontologies (syntax and semantics), combining relational databases with rule-based ontologies, ontological query answering (OQA), universal models, ontology-based data access; (5) Ontological query answering: forward-chaining (the chase procedure), backward-chaining (resolution-based query rewriting), linear rule-based ontologies (tractable data complexity, intractable combined complexity); (6) Advanced topics (time permitting): expressive rule-based ontology languages, chase termination, static analysis of ontological queries.
Prerequisites: While there are no formal prerequisites, it is recommended that students have passed a Discrete Mathematics course, and an introductory course in Databases (some familiarity with the relational model, and the main relational query languages). It is also recommended that students have some basic familiarity with computational logic (first-order logic), and complexity theory (standard complexity classes such as PTIME and NP).
Bibliography:

1. S. Abiteboul, R. Hull, V. Vianu, *Foundations of Databases*, 1995.
2. M. Arenas, P. Barcelo, L. Libkin, W. Martens, A. Pieris, *Principles of Databases* (currently under development, a preliminary version is accessible at https://github.com/pdm-book/community)

3. F. Baader, I. Horrocks, C. Lutz, U. Sattler, *An Introduction to Description Logic*, 2017.
4. L. Libkin, *Elements of Finite Model Theory*, 2012.

Teaching methods: Lectures (3 hours weekly), Recitation (1 hour weekly).
Assessment: Coursework (20%), essays (50%), in-class presentations (30%).

MAI650 Internet of Things
Type: Restricted choice for MAI
Level: Graduate
Semester: Spring
Credit: 8 ECTS units
Instructor: V. Vassiliou
Language: English
Objectives: The internet of things (IoT) is a computing concept that describes the idea of everyday physical objects being connected to the internet and being able to identify themselves to other devices. IoT has emerged as a new paradigm aimed at providing solutions for integration, communication, data consumption, and analysis of smart devices. To this end, connectivity, interoperability, and integration are inevitable parts of IoT communication systems. Whereas IoT, due to its highly distributed and heterogeneous nature, is comprised of many different components and aspects, providing solutions to integrate this environment and hide its complexity from the user side is inevitable. In this course, different building blocks of IoT, such as sensors and smart devices, M2M communication, data collection and processing and the role of humans in future IoT scenarios are elaborated upon and investigated. The major focus will be to provide an overview on IoT tools and applications and to introduce hands-on IoT communication concepts through lab exercises.
Content: Upon completion of this course, the student should be able to: (1) Explain the definition and usage of the term "Internet of Things" in different contexts; (2) Understand and describe the key components that make up an IoT system; (3) Apply the knowledge and skills acquired during the course to build and test a complete, working IoT system involving prototyping, programming and data analysis; (4) Independently research the technological trends which have led to IoT; (5) Understand where the IoT concept fits within the broader ICT industry and recognize possible future trends; (6) Valuate the impact of IoT on society by analyzing IoT systems with regard to sustainability, safety, integrity and ethics; (7) Appreciate the role of big data, cloud computing and data analytics in a typical IoT system.
Prerequisites: ----------
Bibliography:

1. Rajkumar Buyya, Amir Vahid Dastjerdi, *Internet of Things Principles and Paradigms*, Morgan Kaufmann, 1st Edition, 2016.
2. J. Biron, J. Follett, *Foundational Elements of an IoT Solution*, O'Reilly Media, 2016.
3. Jamil Y. Khan, Mehmet R. Yuce, *Internet of Things (IoT) Systems and Applications*, 2019.
4. David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Robert Barton, Jerome Henry, *IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things*, Cisco Press, 2016.

Teaching methods: The course is taught through: (1) Interactive face-to-face lectures; (2) Group activities/discussions; (3) In class/lab activities; (4) Student Presentations; (5) Guest Lectures or significant recorded public lectures.
Assessment: Midterm exam (20%), assignments/project (30%), final exam (50%).

MAI651 AI and Creativity
Type: Restricted choice for MAI
Level: Graduate
Semester: Fall
Credit: 8 ECTS units
Instructor: Special Teaching Scientist

Language: English
Objectives: Artificial intelligence (AI) is typically known as a collection of tools for solving hard problems spanning areas such as computer vision, speech and natural language processing. A less known use of AI is its capability for creativity. Through this course, the students will become familiar on how to build creative AI systems, and how various types of AI can be used for creative exploration in art, music and design.
Content: Upon completing this course, the students are expected to: (1) Understand how AI can be creative; (2) Implement evolutionary algorithms for open-ended exploration; (3) Implement generative deep learning models and explore their latent space; (4) Know how to use AI tools as a creative aid in a variety of applications; (5) Understand how computer games and AI can work together to find creative solutions for very hard real-worldproblems; (6) Become familiar with the cultural and creative industry and follow a series of invited lectures of distinguished individuals with practical examples of their work at the intersection of arts, science and technology.
Prerequisites: MAI611 AI Fundamentals, MAI612 Machine Learning.
Bibliography:

1. Bentley, P. J., Corne, D. W., *Creative evolutionary systems*, Morgan Kaufmann, 2002.
2. Floreano, D., Mattiussi, C., *Bio-inspired artificial intelligence: theories, methods, and technologies*, MIT Press, 2008.
3. Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y, *Deep learning*, MIT press, 2016.
4. McCormack, J., d'Inverno, M. (eds.), *Computers and Creativity*, Springer, 2012.
5. Pereira, F. C., *Creativity and Artificial Intelligence: A Conceptual Blending Approach*, Applications of Cognitive Linguistics series, Mouton de Gruyter, 2007.
6. Veale, T, *Exploding the Creativity Myth: The Computational Foundations of Linguistic Creativity* Bloomsbury Academic, 2012.
7. Veale, T., Feyaerts, K., Forceville, *Creativity and the Agile Mind: A Multidisciplinary study of a Multifaceted phenomenon*, Mouton de Gruyter, 2013.

Teaching methods: Teaching will be done through lectures given physically.
Assessment: Assignments (20%), Project presentations (40%), Final exam (40%).

Short Biographical Notes of Academic Staff

Elias Athanasopoulos, Associate professor. Before joining UCY he was an assistant professor with Vrije Universiteit Amsterdam. He holds a BSc in Physics from the University of Athens and a PhD in Computer Science from the University of Crete. He is a Microsoft Research PhD Scholar. He has interned with Microsoft Research in Cambridge and he has worked as a research assistant with FORTH in Greece from 2005 to 2011. He is also a Marie Curie fellow. Before joining the faculty of Vrije Universiteit Amsterdam, he was a postdoctoral research scientist with Columbia University and a collaborating researcher with FORTH. He has several publications in the top security conferences like IEEE Security and Privacy, ACM CCS, Usenix Security, and NDSS.

Andreas Aristidou, Assistant professor. He had been a Cambridge European Trust fellow, at the Signal Processing and Communications Laboratory, University of Cambridge, where he obtained his PhD in Information Engineering (2011). He has a BSc in Informatics and Telecommunications from the National and Kapodistrian University of Athens (2005) and he is an honor graduate of Kings College London (2006), where he obtained his MSc degree in Mobile and Personal Communications (2006). Previously, he served as Senior PostDoc Researcher at the Interdisciplinary Center Herzliya in Israel, the Shandong University in China, the University of Cyprus, and the Cyprus University of Technology. He has also taught as a Visiting Lecturer at the University of Nicosia. He is a senior member of the Institute of Electrical and Electronic Engineers (IEEE), the Association of Computing Machinery (ACM), Eurographis, and the Cyprus Scientific and Technical Chamber, where he served as a member of the Research Committee. Andreas is on the editorial board of The Visual Computer (TVC) journal, and he is guest editor of the Advances in Applied Clifford Algebras (AACA) journal. His main research interests are focused on 3D motion analysis and classification, motion portrayal and synthesis using deep learning techniques, human animation in games and simulations, including intangible cultural heritage, and involve motion capture, inverse kinematics, and applications of conformal geometric algebra in graphics.

Chris Christodoulou, Professor. Undergraduate studies at Queen Mary and Westfield College, University of London, UK (BEng in Electronic Engineering, 1991) and Birkbeck College, University of London, UK (BA in German, 2008). Graduate studies at the Kings College, University of London, UK (Ph.D. in Electronic Engineering, 1997). He has worked as a Postdoctoral Research Associate at the Kings College, University of London, UK (1995-97), and he has taught as a Lecturer (Assistant Professor) at Birkbeck College, University of London, UK (1997-2005). His current research interests include computational neuroscience, neural networks and machine learning.

Yiorgos Chrysanthou, Professor. Head of the Graphics and Hypermedia lab and Research Director of the Centre on Interactive Media, Smart Systems and emerging Technologies (RISE). Educated in the UK (BSc and PhD from Queen Mary and Westfield College) and worked for several years as a research fellow and a lecturer at University College London. He has published over 80 papers in journals and international conferences on computer graphics and virtual reality and is a co-author of the book "Computer Graphics and Virtual Environments: From Realism to Real-Time", (Addison-Wesley 2001+ China Machine Press 2004). Yiorgos serves as an associate editor for the Journals Computer Graphics Forum and Computers and Graphics, and review editor for Frontiers in Robotics and AI (Specialty Section Virtual Environments). He served as the local or overall coordinator of over 25 research projects, related to 3D graphics, virtual reality and applications and as Organising or Program Chair for several conferences. His research interests lie in the general area of 3D Computer Graphics, recently focusing more on the development of algorithms for real-time

AR and VR rendering, reconstruction of urban environments and computer animation.

Marios D. Dikaiakos, Professor. Undergraduate studies at the National Technical University of Athens (Dipl.-Ing. In Electrical Engineering, summa cum laude, 1988). Graduate studies at Princeton University (Ph.D. 1994, M.A. 1991 - both in Computer Science). He has worked and taught at the University of Washington, Seattle, USA (1994-1995) and the University of Cyprus (1996; 1998-today). He also taught as visiting professor at the University of Crete, Greece (10/2004), Rutgers University, USA (spring 2005), National Technical University of Athens, Greece (fall 2021) and Université de Paris Cité (5/2022). At the University of Cyprus, professor Dikaiakos has served as Head of the Computer Science Department (2010-2014) and founding Director of the Centre for Entrepreneurship (2014-2021). He also established and is the Principal Investigator of the Laboratory for Internet Computing at the Department of Computer Science, University of Cyprus, where he directs several research projects funded by the European Union and the Research and Innovation Foundation of Cyprus. Prof. Dikaiakos is Associate Editor of Springer's Computing jornal, has served in the program committees of many international scientific conferences and has consulted with the European Commission and several organizations of European countries and China in evaluating research proposals, research projects, and academic work. Prof. Dikaiakos has published over 160 papers in peer-reviewed scientific journals and academic conferences. His research interests include Network-Centric Computing with an emphasis on Cloud Computing systems (Performance, Dependability, Monitoring) and Online Social Network Analysis and Knowledge Extraction.

Yannis Dimopoulos, Professor. Undergraduate and graduate studies at the Athens University of Economics (B.Sc. in Computer Science 1987, Ph.D. in Computer Science, 1992). He has worked as Researcher at the Max-Planck-Institute for Computer Science, Germany (1992-1994), University of Cyprus (1995-1996) and University of Freiburg, Germany (1996-1998). His research interests are in the area of Artificial Intelligence with emphasis on Knowledge Representation and Reasoning, Planning, Logic Programming, Constraint Satisfaction and Machine Learning.

Chryssis Georgiou, Professor. Undergraduate studies at the University of Cyprus, Cyprus (B.Sc. in Mathematics, 1998). Graduate studies at the University of Connecticut, USA (M.Sc., 2002; Ph.D., 2003, both in Computer Science and Engineering). Before joining the University of Cyprus in 2004, he worked as a Teaching and Research Assistant at the University of Connecticut, USA (1998-2003). His research interests span the Theory and Practice of Fault-tolerant Distributed and Parallel Computing with a focus on Algorithms and Complexity. He has published more than 100 articles in journals and conference proceedings in his area of study and he has coauthored two books on Robust Distributed Cooperative Computing. Recently, he co-edited a book on the Principles of Blockchain Systems. He has served on several Program Committees of conferences in Distributed and Parallel Computing and on the Steering Committees of the International Symposium on Distributed Computing-DISC (2008-2010, 2010-2012) and the ACM Symposium on the Principles of Distributed Computing-PODC (2014-2015). Currently he is serving a three-year term (2021-2024) as the Chair of the Steeting Committee of PODC. In 2015 he served as the General Chair of PODC 2015, in 2017 he served as the Track Program Committee co-Chair (Stabilizing Systems: Theory and Practice Track) of SSS 2017, and in 2018 he served as the General co-Chair and PC co-Chair of the workshop ApPLIED 2018 (co-located with PODC 2018). In 2019, 2021 and 2022 he served as the General co-Chair of ApPLIED 2019, 2021, and 2022 (co-located with DISC 2019, PODC 2021, and PODC 2022, respectively). In 2020 he served as the PC co-Chair of NETYS 2020. As of January 2018, he is on the Editorial Board of Information Processing Letters (subject area: Distributed Computing). Dr. Georgiou's research has been funded by the University of Cyprus, the Cyprus Research and Innovation Foundation, and the European Commission.

Antonis C. Kakas, Professor. Undergraduate studies at the Imperial College, University of

London, UK (B.Sc. in Mathematics, 1980). Graduate studies at the Imperial College, University of London, UK (M.Sc. in Computer Science, 1987 and Ph.D. in Theoretical Physics, 1984). He has worked as Research Fellow at Kings College, UK (1984-1986), at the University of Zurich, Switzerland (1987-1988), and at Imperial College, UK (1988-1992). He is a member of the editorial board of the international journals *AI Communications, Journal of Applied Logic* and *Journal of Theory and Applications of Logic Programming.* His research interests include Knowledge Representation and Reasoning in Artificial Intelligence, with applications to Machine Learning, Planning, Diagnosis, Cognitive Agents Computational Bioscience.

Georgia (Zeta) Kapitsaki, Associate Professor. Undergraduate studies at the National Technical University of Athens (Dipl. Electrical and Computer Engineering, 2005). Graduate studies at the National Technical University of Athens (M.Sc. in Technoeconomical Systems, 2008), and Ph.D., 2009, School of Electrical and Computer Engineering. She has worked as a Research Associate at the National Technical University of Athens (2005-2009), as a Laboratory Assistant at the Technical Institute of Piraeus (2007-2009), as well as a software and telecommunications engineer in Germany (2005, 2009-2010). She has also worked as a visiting researcher at the Otto von Guericke University of Magdeburg and TU Delft. She has been involved in the organisation of international conferences (e.g. ICSME 2022, SAC 2019, ICSR 2016, etc.). She has been serving as a member of the program committee of international conferences (e.g., WISE 2022, ICWS 2022, MSR 2022, ENASE 2021, WEBIST 2021). She has published over 60 papers in international journals and conferences, and is contributing to open source projects. She has received the best paper award in ICSR 2015 and in the doctoral symposium of MODELS 2008. She has served as an evaluator for EU proposals. She has been a Mentor in Google Summer of Code 2018 and 2019. She is the Representative of the University of Cyprus at GFOSS Open Technologies Alliance. She has been involved and received funding from European research projects (e.g. SocioCoast, CYberSafety, TAMIT). She is a member of the editorial board of ERCIM News of the journal of Applied Computing and Intelligence, and has been a guest editor for Elsevier Journal of Systems and Software. Her research interests include Software Engineering, Open Source Software, Human Aspects in Software Engineering, Context-aware applications and Privacy Enhancing Technologies.

Elpida Keravnou-Papailiou, Professor. Undergraduate studies at Brunel University, UK (B. Tech. in Computer Science, 1982). Graduate studies at Brunel University, UK (Ph.D. in Cybernetics, 1985). She has taught at University College London, UK (Lecturer, 1985-1991; Senior Lecturer, 1991-1992; Director of the M.Sc. Program in Computer Science). She served as first Rector of the Cyprus University of Technology (2012-2015), as Chair of the Governing Board of the Cyprus University of Technology (2009-2010), as Chair of the Cyprus Council for the Recognition of Degrees (2004-2008), as Vice Chair of the Evaluation Committee for Private Universities (2005-2015), as a Member of the Council for Educational Evaluation and Accreditation (2001-2006), as a Member of the Governing Board of the Cyprus National Scholarships Foundation (2003-2006), as a Member of a Committee of Specialists for Education Reforms in Cyprus (2003-2004), as a Member of the Preparatory Committee for the establishment of the Cyprus Open University, and as a Bologna Expert and contact point for academic recognition in Cyprus. At the University of Cyprus she served as Vice Rector for Academic Affairs (2002-2006), as Dean of the School of Pure and Applied Sciences (1999-2002), and as Chair of the Department of Computer Science (1994-1998) where she is currently serving again as Chair (2016-2018). During her term as Vice Rector for Academic Affairs she coordinated the implementation of the Bologna reforms at the University of Cyprus. She is Associate Editor of the journal *Artificial Intelligence in Medicine* (Elsevier) since the launching of the journal in 1989, and a Member of the Editorial Board of the journal *Systems Research and Information Science* (Gordon and Breach) and of the electronic open access journal *Advances in Artificial Intelligence* (Hindawi Publishing Corporation – www.hindawi.com/journals/aai). During the period 2003-2005 she served as

Chair of the Artificial Intelligence in Medicine Europe (AIME) Board. Her research interests include Knowledge Engineering, Artificial Intelligence in Medicine, Expert Systems, Deep Knowledge Models, Diagnostic Reasoning, Temporal Information Systems in Medicine, Temporal Reasoning and Temporal Data Abstraction, Hybrid Decision Support Systems, Quality Assurance, and Flexible Learning Paths. In addition to her publications in scientific journals and conference proceedings she has (co)authored seven scientific books published by Springer, Elsevier, Kluwer, Kogan-Page, Chapman and Hall, Abacus Press, and the Greek Open University. She has served as a Member of the Governing Board (June 2012 – June 2018) and of the Executive Committee (July 2014 – June 2018) of the European Institute of Innovation and Technology (EIT - www.eit.europa.eu).

Marios Mavronicolas, Professor. Undergraduate studies at the National Technical University of Athens, Greece (Diploma in Electrical Engineering, 1985; Summa cum Laude). Graduate studies at Harvard University, USA (M.A., 1988; Ph.D., 1992, both in Computer Science). He has taught at University of Crete, Greece as Visiting Assistant Professor (1992-93), and at the University of Connecticut, USA as Assistant Professor (1999). He is on the Editorial Board of the journals *Theoretical Computer Science, Journal of Interconnection Networks* and *Networks*, and he is the Distributed Computing Column Editor of the *Bulletin of the European Association for Theoretical Computer Science*. He has been the Program Committee Chair for the 11th International Workshop on Distributed Algorithms (WDAG 1997), the Program Committee Co-Chair for the 2nd International Workshop on Internet and Network Economics (WINE 2006), and the Program Committee Chair for the 2nd International Symposium on Algorithmic Game Theory (SAGT 2009). He is the Chairman of the Cyprus Council for the Evaluation of Higher Degrees (ΚΥ.Σ.Α.Τ.Σ.) and the Vice-Chair of the Assembly of Rectors of Cypriot Universities. His research interests span the area of Algorithms and Complexity, with emphasis on Algorithmic Game Theory, Distributed and Parallel Computing, Networking and the Internet.

George Pallis, Associate Professor and Associate Director of Laboratory of Internet Computing. BSc (2001) and Ph.D. (2006) degree in Department of Informatics of Aristotle University of Thessaloniki (Greece). His research interests include Cloud computing with focus on Cloud elasticity and monitoring, Edge Computing and Big Data Analytics. Dr. Pallis has published over 70 papers in international journals (e.g., IEEE TKDE, IEEE TCC, IEEE TSC, ACM TOIT etc), magazines (e.g., CACM, IEEE Internet Computing) and conferences (e.g., INFOCOM, IPDPS, ICDCS, IEEE BIG DATA etc) and he is contributor of two international DIN (German Institute for Standardization) standards. His research has attracted more than 3.7M euro (funded by EC, Research Promotion Foundation in Cyprus, and industry (e.g., Google)). Dr. Pallis has served as PC Co-chair of CloudCom 2018 and CCGrid 2019. Dr. Pallis has also served in numerous Program and Organization Committees for international conferences and he received the best paper awards in the IEEE Big Data Conference (IEEE BIG DATA 2016) and the International Conference on Service-Oriented Computing (ICSOC 2014). He is Editor in Chief in the IEEE Internet Computing magazine, Associate Editor in the IEEE Transactions on Cloud Computing and Associate Editor in the Computing Journal.

George A. Papadopoulos, Professor. Undergraduate studies at Aston University, UK (B.Sc. in Computer Science and Mathematics, 1982). Graduate studies at Aston University, UK (M.Sc. in Computer Science, 1983) and at the University of East Anglia, UK (Ph.D. in Computer Science, 1989). He has worked as a Researcher at NCR Demokritos, Greece (1981-1982), as a Research Assistant at the Aristotelian University of Thessaloniki, Greece (1983-1985) and as a Research Associate at the University of East Anglia, UK (1987-1990). He is a recipient of an 1995 ERCIM Fellowship visiting GMD in Germany and CWI in the Netherlands. His research interests include Advanced Software Engineering, Parallel and Distributed Systems, Adaptive and Context Aware Systems, e-Health and e-Learning.

Constantinos S. Pattichis. Professor. Undergraduate studies at the Higher Technical

Institute (H.N.D. in Electrical Engineering, 1979), and at the University of New Brunswick, Canada (B.Sc. in Electrical Engineering, 1983). Graduate studies at the University of Texas at Austin, USA (M.Sc. in Biomedical Engineering, 1984), at the University of Newcastle Upon Tyne, UK (M.Sc. in Neurology, 1988), and at the Queen Mary College, University of London, UK (Ph.D. in Electronic Engineering, 1992). He has worked as a Research Associate (1986-87) and as a Senior Research Scientist (1990-92) at the Cyprus Institute of Neurology and Genetics and has taught as Visiting Lecturer at the Higher Technical Institute of Cyprus (1990-1992). He is Associate Editor of the scientific journals *IEEE Transactions on Information Technology in Biomedicine* and the *IEEE Transactions on Neural Networks*. He served as Chairman of the IEEE Cyprus Section, the Cyprus Association of Medical Physics and Biomedical Engineering and of the conferences Medical and Biological Engineering and Computing (1998) and IEEE MELECON 2000. His research interests include the development of Computational Intelligent Systems (based on Neural Networks and Genetic Algorithms), Data Mining in Medicine, Medical Imaging Analysis and Biosignal Analysis, and Health Telematics.

Anna Philippou, Professor. Undergraduate studies at the University of Oxford, UK (B.A. in Mathematics and Computation, 1992). Graduate studies at the University of Warwick, UK (M.Sc. in Parallel Computers and Computation, 1993; PhD in Computer Science, 1997). She has worked as a Teaching Assistant at the University of Warwick, UK (1993-1996) and as a Postdoctoral Research Fellow at the University of Pennsylvania, USA (1997-1998). Her research interests include Concurrency Theory and its Applications, Specification and Verification of Concurrent Systems, Formal Methods for Safety-Critical Systems and Algorithmic Game Theory. She has published in top journals and international conference proceedings in her research field. She has also served in the Program Committees of the top conferences on Formal Methods, co-chaired the Program Committee of the TACAS 2009 Conference, and served as the General Chair of the ETAPS 2010 and FM 2016 conferences. Her research activity has been funded by the Research Promotion Foundation and the European Commission.

Andreas Pieris, Assistant Professor. Undergraduate studies at the University of Cyprus (B.Sc. in Computer Science, 2006). Graduate studies at the University of Oxford (M.Sc. in Mathematics and Foundations of Computer Science, 2007; D.Phil. in Computer Science, 2011). He has worked at the University of Edinburgh (Associate Professor since 2020; Assistant Professor, 2016 - 2020), the Vienna University of Technology (Postdoctoral Researcher, 2014 - 2016), and the University of Oxford (Postdoctoral Researcher, 2011 - 2014). His research interests are database theory with emphasis on uncertain data, knowledge representation and reasoning, and logic in computer science. He has published numerous papers in leading international conferences and journals. He has also served on the program committees of several conferences, including the top-tier database theory and artificial intelligence conferences.

Andreas Pitsillides, Undergraduate studies at University of Manchester, Institute of Science and Technology (UMIST), UK (B.Sc. in Electrical and Electronic Engineering, 1980). Graduate studies at Swinburne University of Technology, Australia (Ph.D. in High Speed Multimedia Networks, 1993). Andreas co-directs the Networks Research Laboratory (NetRL, http://www.NetRL.cs.ucy.ac.cy), which he founded in 2002, and is appointed Visiting Professor at the Department of Electrical & Electronic Engineering Science, University of Johannesburg (2021-2024). Prior to that he held a Visiting Professor appointment at the University of the Witwatersrand (Wits), School of Electrical and Information engineering, Johannesburg, South Africa (2017-2021) , and the University of Johannesburg, Department of Electrical and Electronic Engineering Science, South Africa (2014 -2017). Andreas was the Department of Computer Science Chair from 2014-2016, a Founding member and Chairman of the Board of the Cyprus Academic and Research Network (CYNET) from 2000-2016. His broad research interests include communication

networks (fixed and mobile/wireless), Nanonetworks and Software Defined Metasurfaces and Metamaterials, Programmable Wireless Environments, the Internet of Things, Smart Systems (e.g. Smart Grid) and Smart Spaces (e.g. Home, City), and Internet technologies and their application in Mobile e-Services, especially e-health, and security. He has a particular interest in adapting tools from various fields of applied mathematics such as adaptive non-linear control theory, computational intelligence, game theory, and recently complex systems and nature inspired techniques, to solve problems in communication networks. Published over 350 referred papers in flagship journals (e.g. IEEE, Elsevier, IFAC, Springer), international conferences, and books and book chapters, participated in over 40 European Commission and locally funded research projects with over 7 million Euro as principal or co-principal investigator, received several awards, including best paper, presented keynotes, invited lectures at major research organisations, short courses at international conferences and short courses to industry. He serves on the editorial board of the ITU Journal on Future and Evolving Technologies (ITU J-FET), served on the editorial boards of the Journal of Computer Networks (COMNET), serves/served on many international conferences as general chair, technical program chair, and on executive committees, technical committees, guest co-editor, invited speaker, and as a regular reviewer for conference and journal submissions.

Yanos Sazeides, Assistant Professor. Undergraduate studies in Computer Engineering at Oakland University (B.Sc.,1991). Graduate studies at Cornell University (M. Eng. in Electrical Engineering, 1992) and at University of Wisconsin-Madison (Ph. D. in Electrical Engineering, 1999). He has worked at research/development laboratories for high performance processors at HP, Compaq and Intel. He has taught at the University of Cyprus as Visiting Lecturer (2000-2001). His research interests include Computer Architecture with special emphasis on High Performance Processors, Program Behavior, Execution Paradigms and Power Aware Microarchitectures.

Vasos Vassiliou, Associate Professor. Undergraduate studies at the Higher Technical Institute (H.N.D. in Electrical Engineering, 1993) and the University of South Florida (B.Sc. in Electrical Engineering, 1997). Graduate studies at the Georgia Institute of Technology, USA (M.Sc., 1999 and Ph.D., 2002, both in Electrical and Computer Engineering). He has worked as Research and Teaching Assistant at the Georgia Institute of Technology, USA. He has taught as a Visiting Lecturer at the Department of Computer Science, University of Cyprus (2004-2005) and as an Assistant Professor at the Intercollege, Cyprus (2002- 2004). Is the chairman of the Board of Directors of the Cyprus Research and Academic network (CYNET) since 2016. Είναι Πρόεδρος του Διοικητικού Συμβουλίου του Κυπριακού Ερευνητικού και Ακαδημαϊκού Δικτύου (ΚΕΑΔ). He is a member of the Institute of Electrical and Electronic Engineers (IEEE) and of the Association of Computing Machinery (ACM). He is a of the IEEE Cyprus Section Executive Committee and Chair of the IEEE Communications Society Cyprus Chapter. His research interests include High Speed Network Architectures, Mobile Computing, Wireless Communications, Quality of Service (QoS) and Traffic Engineering for Computer and Telecommunication Networks.

Haris Volos, Assistant professor. Undergraduate studies at the National Technical University of Athens (Dipl.-Ing. in Electrical and Computer Engineering, 2005). Graduate studies at the University of Wisconsin-Madison (M.Sc., 2007 and Ph.D., 2012, both in Computer Sciences). He has worked as a researcher at Hewlett Packard Labs, Palo Alto (2013-2018) and as a software engineer at Google, Mountain View (2018-2019). His research interests span broad areas of computer systems, including computer architecture, operating systems, and data storage and processing.

Demetris Zeinalipour, Associate Professor. Undergraduate studies at the University of Cyprus, Cyprus (B.Sc. in Computer Science, 2000). Graduate studies at the University of California - Riverside, USA (M.Sc., 2003; Ph.D., 2005, both in Computer Science and Engineering). He has worked at the University of California - Riverside, CA, USA (Teaching

and Research Assistant, 2000-2005), the University of Cyprus (Visiting Lecturer, 2005-2007), the Open University of Cyprus (Lecturer, 2007-2009) and the University of Cyprus (Lecturer, 2009-2012 and Assistant Professor, 2013-now). He has also a short-term visiting researcher at Akamai Technologies, Cambridge, MA, USA (2004), a Marie-Curie Fellow at the University of Athens, Greece (2007) and a visiting researcher at the University of Pittsburgh, PA, USA (2015). During 2016-2017, he was on a sabbatical at the Max Planck Institute for Informatics, Saarbrücken, Germany. He has founded and directs the Data Management Systems Laboratory (DMSL) at the Department of Computer Science at the University of Cyprus. He is a Senior Member of Association for Computing Machinery (ACM), New York, NY, USA, a Senior Member of Institute of Electrical and Electronics Engineers (IEEE), Piscataway, NJ, USA, a Member of the Advanced Computing Systems Association (USENIX), Berkeley, CA, USA and a Humboldt Fellow of the Alexander von Humboldt Foundation, Bonn, Germany. His research interests include Data Management in Computer Systems and Networks, particularly Mobile and Sensor Data Management; Big Data Management in Parallel and Distributed Architectures; Spatio-Temporal Data Management; Network and Telco Data Management; Crowd, Web 2.0 and Indoor Data Management; Data Privacy Management.

Appendix A: Rules for Diploma Projects

1. GENERAL

A *Diploma Project* is prepared by a final-year student, usually during the seventh and eighth semesters of study in accordance with the Programme of Studies of the Department. The part of Diploma Project prepared in the first semester is called *Diploma Project I* and the part corresponding to the second semester *Diploma Project II*. These two parts are prepared and completed in *EPL400 - Diploma Project I* and *EPL401 – Diploma Project II*, respectively.

A Diploma Project corresponds to seventeen and a half (17.5) *Credit Units ECTS*, which are credited to the student upon successful completion.

The student registers for the Diploma Project with the approval of the Academic Advisor.

The Department Council appoints a member of the Faculty of the Department, who ensures and coordinates the entire process of developing and assessing the Diploma Projects. This Faculty member is known as the *Diploma Projects Coordinator*.

All the forms and guides mentioned in this section are digitally available on the website of the Department. The Department maintains a *Digital Library for the Diploma Projects* for archiving purposes.

2. SUBSTANCE, FORM AND EVALUATION CRITERIA

2.1 Substance

Each *Diploma Project* must contain sufficient information that reflects the student's initiative, independent study and productivity (originality, in the broad sense).

The Diploma Project may be of theoretical or practical nature or a combination of both. A Diploma Project may include, for example, an application of existing techniques, extension of known methods in the theory, software, hardware or applications areas, development of a prototype system, address theoretical problems, a survey review or study of a theoretical or practical area, etc.

2.2. Form

A *Diploma Project* must be a comprehensive document structured in chapters and must follow the rules of the technical guidance report called *Standards for Preparation the Diploma Project*.

The Diploma Project should include an introduction to the subject, an analysis of the importance of project, a description of the related work, a review the work in the area of the topic, a description of the methodology used, listing, classification and evaluation of the results of the work and finally conclusions and suggestions for possible extension of the work.

Diploma Projects where software was created and/or used should also include the code of the software in a specific Annex, a description and analysis of the software a separate section, and instructions for the use of the software. The code of such software may not be used as the Thesis document.

2.3 Evaluation Criteria

The main criteria for the evaluation of the Diploma Project are the following:

[a] Quality of work (e.g. accuracy and completeness of analysis, appropriateness of methodology, validity of theoretical results, software quality, implementation, consistency of material presented and association of ideas).

[b] Degree of the objectives achieved of the Diploma Project.
[c] Degree of understanding by students of the area of the Diploma Project topic.
[d] Quality of the written language of Diploma Project (e.g. structure and organization, clarity, ease of reading and understanding).
[e] Quality of Presentation of the Diploma Project. (e.g. oral speech, the adequacy and suitability of multimedia used (such as slides), proper utilization of the allowed time of presentation, and most importantly to point the contribution of Diploma Project through the presentation).

3. PROCEDURE

3.1 Preparation

3.1.1 Submission and Announcement of Topics

Each member of the Faculty submits to the Department, end of March of each year, a number of Diploma Project topics greater than the ratio of the number of students per Faculty member. Each Faculty member reserves the right not to supervise a number of Diploma Projects larger than the nearest integer above the ratio of students per Faculty member.

Each topic has a title. It is expected that a brief description of each topic is made available to the students by the proposing faculty. (It is also expected that the faculty members will update and renew appropriately the description of the projects on their personal websites).

The *Diploma Projects Coordinator establishes a list of the submitted Diploma Project topics*, and announces the Diploma Project list to the faculty members of the Department and to the final-year students approximately in middle of April.

3.1.2 Choice of Topic

The successful completion of at least 156 ECTS is a necessary prerequisite for the assignment of a Diploma Project.

Each student should choose a Diploma Project topic. To this end, the students shall discuss with the corresponding faculty members that have offered the topics, in meetings during a specified period of time. Throughout the process of submission of preferred topics, the students obtain the agreement of the faculty member for the supervision of the corresponding project. This Faculty member will be the *Diploma Project Supervisor* for that student.

With the selected choice of the topics, the student completes a special *Registration Form*, available electronically on the website of the Department which includes the title, description, and any specialized software / hardware or other resources necessary for the preparation of the Diploma Project. The *Registration Form* is signed by the Faculty member, the Academic advisor of the student and it is deposited to the Department no later than the period of registration of students for the semester the student prepare the *first part of the Diploma Project*.

In case of not being able to select topic, the student returns the *Registration Form* to the Department, signed by him/her and the Academic Advisor. In this case, with the filing of the form, the *Diploma Project Coordinator* contacts the Faculty members of the Department, to ensure that every student who submitted the registration form without topic to select a topic before the end of the submission period. In case this is not possible, a special meeting of the Department Council is called to assure that each student is given a topic. The student must choose the topic selected by the Council meeting

Concurrently with the submission of the registration form, the student must register for the EPL 400 course.

3.1.3 Change of Subject

Changing topic of the Diploma Project (with the same or another supervisor) is possible within the first three (3) weeks of the semester the student enrolled for the *first part of the Diploma Project* (Diploma

Project I). To this end, using the same procedure that was followed for the submission of the original Diploma Project topic, the student fills in the *Special Registration* Form to the Department in order for the *coordinator of Diploma Project* to approve it.

3.2 Preparation

3.2.1 Supervision

The supervision of a student who prepares a Diploma Project is the responsibility of the Diploma Project Supervisor. The monitoring and controlling of the Diploma Project progress is done through regular meetings between the student and the supervisor. In some cases a co-supervisor may be indicated. The supervisor must be a PhD holder and may either come from the Department of Computer Science or from an institute or organization outside the Department.

3.2.2 Interim Evaluation

During the examination period of the first semester in which the student enrolled in the Diploma Project, the student submits to the *Supervisor* a brief Progress Report. Following the submission of the Progress Report, the Supervisor shall submit a written assessment of the progress of the student to the Department, which is also sent to the student. The possible grades for this assessment are Success or Failure. The grade of the Diploma Project I is the student's grade for the course EPL400 which has been enrolled for the current semester.

In the second semester the student can enroll in the Diploma Project II, EPL401, only if the student has succeeded with EPL400 course. In case of a Failure grade results in the student enrolling again in the first part of the Diploma Project (Diploma Project I: EPL400) in the same or different topic (with the same or different supervisor).

3.3 Evaluation

The final evaluation of the Diploma Project takes place towards the end of the semester where the student is registered for *Diploma Project II*.

3.3.1 Second Assessor

The Diploma Project is assessed by the *supervisor*, and the co-supervisor (if assigned), together with another member of the Faculty, known as the *Second Assessor*. The Diploma Project Coordinator with the collaboration of the Supervisors, publishes a *list of Assessors and Program of Diploma Project presentations*.

In special cases, the Coordinator of the Diploma Project may approve as a *Second Assessor* a Visiting Faculty Member of the Department or a Member of the Special Teaching Staff of the Department or a Faculty member from other Department of the University or/and other University in Cyprus or abroad. The approval may be justified in cases of close affinity of the Second Assessor Diploma on the subject of the Diploma Project.

3.3.2 Presentations

The Diploma Project Coordinator publishes the *Program of Diploma Project presentations* for three days during the week immediately following the examination period. The Program of Diploma Project presentations must be made publicly available to all faculty members of the department and the presentations are open to the public.

Each student who is expected to complete its Diploma Project II based on the Academic Advisor judgment must be listed in the *Diploma Project Presentation list*. These students present their Diploma Project in public in front of their *Academic Advisor* and their *Second Assessor*.

The students who have been excluded from the *Program of the Presentations* of the diploma Project receive a grade equal to *CD (Continuation of Diploma Project)* and must continue working on the same Diploma Project so in the immediate following semester to complete their Diploma with success. Therefore, these students must reregister in the same course, *EPL401*.

3.3.3 Grading

After the presentation the *Supervisor*, in consultation with the *Second Assessor*, submits the grade for

the Diploma Project with written justification comments, in a specific *Assessment* Form, and in accordance with the procedure for filing course grades. The Assessment Form of the Diploma Project contains different *evaluation criteria* that need to be filled with numerical grades.

A *Failure* grade results in the student enrolling again in the first part of the Diploma Project (*Diploma Project I*) in a different topic with a different *supervisor*. A Success grade in Diploma Project receives a numerical grade according to the Rules Studies.

The grade of the Diploma Project is the grade for the courses *EPL401* which is enrolled.

The *Coordinator of the Diploma* Project handles the cases where there is a disagreement on the grade for the Diploma Project between the *Supervisor* and the *second assessor*.

3.3.4 Submit in Digital Format

Within ten days after the presentation and evaluation of the Diploma Project, the student shall submit to the Department its Diploma project in digital form. Failure to submit timely may result in delay of the student's graduation which may cause the graduation of the student impossible in the current semester.

3.3.5 Final Grade Submission

The coordinator of the Diploma Project submits the student's grade for the Diploma Project to the Academic Affairs and Student Welfare in time.

Appendix B: Rules for Graduate Studies

The Department of Computer Science offers *Graduate Degree Programs* that lead to *Master in Advanced Information Technologies* and *PhD* in Computer Science. The programs are published at the Department's Guidebook. The *Graduate Program Committee* coordinates the programs based on the General Rules of the University of Cyprus.

[1] Admission Requirements

I. 1. The department announces once a year the maximum number of graduate students it can accept for the upcoming academic year starting in September. Applications arrived after the submission deadline is accepted only in case the number of graduate students is not met.

2. The applicants must submit an application, a CV, official transcripts from all colleges/universities attended, a personal statement about his/her goals and interests, and two letters of recommendations (possibly by faculty members of their college or university, mailed directly to the department). It is not necessary for the applicant to hold an undergrad or college degree at the time he submits his application. However, the applicant must get / have a degree in Computer Science or related field of study before he joins the graduate program. The average grade (GPA) of the applicant should be 6.5/10 (or equivalent to that) and the degree should be from an accredited college / university (as it is defined by the General Rules of University of Cyprus).

3. Applications are examined by the Graduate Program Committee. The Graduate Program Committee holds the right, based on its judgment, to call the applicant for personal interview or ask more information. The Graduate Program Committee writes an evaluation and admission report and submits it to the Department Council for approval. The Department Council holds the right to accept fewer students than it had announced. The department submits its report to the school it belongs to.

II. The department assigns an Academic Advisor to every new graduate student. To a PhD student the department assigns a Research Advisor. The Research Advisor is assigned by the Department Council, after suggestion of the Graduate Program Committee, and an agreement between the student and a faculty member. The Research Advisor watches over the research or any other work of the student and provides any necessary guidance.

III. For students whose undergraduate degree was not related to Computer Science there is a possibility to be asked to take some undergraduate courses offered by the department. The student must pass these undergraduate courses in order to continue the graduate program.

[2] Master Degree

I. To get a Master Degree in Computer Science, each student must complete successfully courses (a total of 60 ECTS units) from the Graduate Program and complete a Master Thesis of 30 ECTS units, under the supervision of a Research Advisor. The assignment of a Research Advisor is based on the rules of the University Senate and is done before the student submits the Thesis Proposal. The student can transfer up to 15 ECTS units from courses completed successfully from similar graduate programs. Theses completed at other graduate schools are not transferable.

II. The Thesis must deal with a research topic or a technical issue. It must be of some original contribution or show a thorough and clear understanding of some special topic. Full-time students normally complete the thesis in 6 months study time. When the student has completed successfully courses totaling to 30 ECTS units, he can submit a Form (signed also by his Research Advisor) stating his Master Thesis topic.

III. The Master Thesis is submitted at the department and defended within the time period decided by the Department Council.

1. When a Master Thesis is submitted the Chair of the Department appoints a Thesis Examination Committee of three members. The head of the committee is the student's

Research Advisor. It is possible a member of this committee to be a faculty member of other department or to not be a faculty member but to hold a PhD Degree or have a reputation in the field of study. The Committee can have also external members who do not hold a faculty position. However, the membership must be approved by the Department Council and the external member must hold a PhD Degree or must have a reputation in the field of study.

2. The Master Thesis is defended in a presentation before the Thesis Examination Committee. The head of the Thesis Examination Committee is responsible for the procedures that must be followed during the defense.

3. The Thesis Examination Committee can accept (even with conditions) or reject a Master Thesis. The Thesis Examination Committee writes its decision in a Master Thesis Evaluation Form and submits it to the Department Council for approval. When the Thesis is accepted the department informs the Student Affairs Office for the graduation procedure of the student. When the thesis is rejected, the student can follow the suggestions of the committee and resubmit it for the first and the last time. The submission and thesis defense procedure is followed from the beginning.

IV. The minimum graduation time to get a Master Degree is three semesters and the maximum graduation time is eight semesters.

[3] PhD Degree

I. The basic requirements to get a PhD Degree are:

1. (a) Successful completion of at least 60 ECTS units of graduate level courses. A student with a Master or equivalent degree is partially or fully exempt from this requirement.

(b) Successful completion of the Comprehensive Examination which must be taken no later than the fifth semester of his/her studies. The student submits to the department a request to take the Comprehensive Examination.

2. The Department assigns a Research Advisor to the student that has completed the ECTS units mentioned above.

3. The structure and the subjects of the Comprehensive Examination are decided by the Department. The Comprehensive Examination is general and attempts to measure the student ability to complete the degree. The topics that are examined are Theory, Software, Hardware, and Applications.

4. The Comprehensive Examination is presented before the Comprehensive Examination committee which is elected by the Postgraduate Committee of the department after it has been requested by the student's academic research supervisor. The formation of the Comprehensive Examination committee must be approved by the Department committee. The academic research advisor of the student is the chair of the Comprehensive Examination committee.

5. The Comprehensive Examination consists of three phases. The student should obtain a passing grade in all of them to ensure success in the examination. The three phases are the following:

i. Submit to the Comprehensive Examination Committee a written literature review on the student's Research area. The review should have the length and the quality of a published survey article and must demonstrate the student's adequate knowledge and understanding of the topic under investigation and the open problems. To prepare the review, the Research Supervisor may give the student an indicative bibliography. The review is graded by the members of the Comprehensive Examination Committee with a Pass / Fail. When a review receives a passing grade, the student may proceed to an oral examination.

ii. The student should make an oral presentation of his/her survey to the committee. The presentation lasts 50-60 minutes, including questions addressed by the committee and the audience. The presentation is open to the public and should be announced to the members of the department and the University.

iii. The oral examination of the student, which is made by the members of committee, is closed to the public. The purpose of the examination is to further investigate whether

the candidate has the skills and capability to conduct Doctoral-level research work in Computer Science.

6. The Comprehensive Examination Committee submits for approval to the department a list of students that have succeeded in the examination.

7. A student that fails to pass the first comprehensive examination must take the examination the next time it is offered. A second fail of the student in the Comprehensive Examination disqualifies the student from the PhD program of the department.

8. Second failure in the comprehensive examination implies exclusion from the PhD candidature in the Department.

II. 1. The Post Graduate Program Committee taking into consideration the suggestions of the Research Supervisor appoints a three member Research Committee for the PhD candidate. The chair of the Research Committee is the Research Supervisor of a student. At most, one member of the Research Committee may be a faculty member of another department.

2. The student is allowed to change his/her Research Supervisor. He/She must submit for approval by the Department Board a request explaining in detail the reasons he/she wishes to change supervisor.

III. 1. When the student has passed the Comprehensive Examination he/she submits a written Doctoral Thesis Proposal to his/her Research Committee. The proposal is also presented orally before the Committee members.

a. The Doctoral Thesis Proposal must have the following structure:
    1. Introduction
    2. Motivation
    3. Problem Statement – Hypothesis
    4. Approach
    5. Roadmap
    6. Related Work
    7. Preliminary Work
    8. Work to be done
    9. Timeline
    10. Future work

b. The Research Committee must examine the Doctoral Thesis Proposal before the end of the next semester. The proposal might be accepted or recommended for resubmission. The final acceptance of the proposal must be given before the start of the seventh year of the student study time. Otherwise, the student case (whether to continue or to terminate his PhD candidacy) is discussed at the Department Council.

IV. 1. Every Doctoral student must submit to the Research Supervisor an annual progress report. This report must be submitted before the end of each academic year.

2.The annual progress report is graded by the Coordinator of the Graduate Studies and the Research advisor of the student with a Pass/Fail. The grade is submitted to Student Affairs and is marked in the student's records to which level the student progress in his/her studies (research, writing).

V. 1. Every PhD candidate completes a Dissertation which must be an original and important contribution to the field of study.

2. The written format of the Dissertation is described in the Department's printed forms. The Dissertation can be submitted no sooner than four semesters before student's admission and completing successfully his Comprehensive Examination.

VI. The student submits to the department six (6) copies of the Dissertation and a request (signed also by his Research Advisor) for the formation of a Dissertation Examination Committee. After this procedure, the graduate Program Committee, with suggestions from the Research Advisor, appoints an Examination Committee (based on the rules of University's Senate). The Graduate Program Committee assures a copy of the dissertation is sent to the members of the Examination Committee. The dissertation is defended by the candidate before the Examination Committee. Note that the three faculty members of the Department are usually

the three members of the Research Committee.

VII. The Chair of the Examination Committee decides the date of the Dissertation Defense. The Defense must take place within three months from the Dissertation submission. All members of the Examination Committee must be present at the Defense. The defense procedure is based on the rules of the University's Senate.

1. The approval of at least four members of the Examination Committee is required to award the degree. In this case, the Examination Committee has the right to request for changes or additions, which considers as necessary. The procedure for monitoring the changes or additions is decided by the Examination Committee and is written clearly in the Evaluation Report.

2. The Examination Committee hands over to the Department Chair a written Evaluation Report of the Dissertation and of the Doctoral Candidate in general, together with its recommendations, in a printed form. The Chair verifies that all the rules and procedures have been correctly followed and hands in the report to the Senate.

VI. A maximum of eight (8) academic years are allowed for the completion of the Doctoral Program.

Approved by the Senate, 149th Meeting, 22/5/2002

Appendix C: Specifications for Preparation of the Master Thesis

Abstract. An abstract is *required*. The body of the abstract may not exceed 400 words in length. Please see the sample abstract page for the format.

Minimum Margins. The minimum acceptable margins for all pages of the thesis and the abstract are 3.8 centimeters (1.5 inches) on left and 2.5 centimeters (1 inch) on the top, bottom, and right.

Paper Requirement. All pages of the thesis must be printed on 21 x 29.7 centimeter (8**.**27 x 11.69 inch) *white* paper. This is the regular A4 paper format.

Font and Point Size. Recommended fonts include Arial, Times New Roman, and Helvetica with a point size of either 11 or 12 (preferably 11).

Printing. Either laser printing or photocopying of high quality is acceptable. Inkjet printing is ***not*** acceptable as it is water soluble. Only *one side* of the paper is to be used for printing. Printing should be consistently clear and dark.

Spacing. The text of the thesis should be *double spaced*. Long quotations, footnotes, appendices, and references may be single spaced.

Color. It is recommended that you ***not*** rely on color to convey information (e.g., use symbols or labels rather than colors to differentiate the lines on a graph, or use stripes and cross-hatching instead of colors to distinguish areas on a map).

Photographs and Graphics. Photographs and graphics in the dissertation should be printed or photocopied directly on the paper as high quality black and white images. Scanned images must print clearly. If color must be used, ***only*** color laser or color photocopy printing is acceptable.

Use of materials copyrighted by others. IMPORTANT: Any material included that goes beyond "fair use" requires written permission of the copyright owner. It may be useful to include these in the thesis as an appendix.

Pagination. Preliminary pages (i.e., the approval page, acknowledgments, table of contents, and the like) are to be numbered consecutively using lower case *Roman numerals*. All pages of the text, appendices (if any), and references must be numbered consecutively using *Arabic* numerals. The page number of the first page of each Chapter must be centered on the bottom of the page. In all other pages, the page number must be placed on the top of the page, aligned to the right.

Landscape pages. The top of a landscape page should be at the left margin and the bottom at the right margin. The page number is to be in the same relative position as on the portrait pages. An easy way to apply page numbers to landscape pages is to run them through the printer twice – once for the text, table, or figure (landscape orientation) and once for the page number (portrait orientation).

Sequence of the main components of the dissertation. The appropriate order of the major sections of the thesis follows: the title page, the abstract, the copyright page (if needed), the approval page, acknowledgments, table of contents, the text, bibliography/references, and appendices (if any).

Bibliography/References. The ACM (http://www.acm.org/pubs) or the IEEE (http://standards.ieee.org/resources) reference style should be followed.

Footnotes and Endnotes No specific rules. The format that is prescribed by your advisory committee should be followed.

* * *

It is recommended that you use your full legal name on the abstract, the title page, the copyright page (if appropriate), and on the approval page. Make certain that your name appears exactly the same way in all places.

Take a moment to check that all pages in all copies of your thesis are accounted for and in the proper order before submitting final copies to the Departmental Secretary. *One hardcopy* and *an electronic version* (in pdf format) of the thesis should be submitted to the Departmental Secretary.

* * *

For students that plan to write their thesis using the Latex authoring language, they can download the style file "ucythesis.sty" that creates the required Master thesis format automatically from http://www.cs.ucy.ac.cy/~chryssis/master-specs.html.

Appendix D: Specifications for Preparation of the Doctoral Dissertation

Abstract. An abstract is *required*. The body of the abstract may not exceed 400 words in length. See the sample abstract page for the format.

Minimum Margins. The minimum acceptable margins for all pages of the dissertation and the abstract are 3.8 centimeters (1.5 inches) on left and 2.5 centimeters (1 inch) on the top, bottom, and right.

Paper Requirement. All pages of the dissertation must be printed on 21 x 29.7 centimeter (8.27 x 11.69 inch) *white* paper that it is *at least 25% cotton and 20lbs. weight*. This is a special A4 paper format that can be obtained at bookstores.

Font and Point Size. Recommended fonts include Arial, Times New Roman, and Helvetica with a point size of either 11 or 12 (preferably 11).

Printing. Either laser printing or photocopying of high quality is acceptable. Inkjet printing is ***not*** acceptable as it is water soluble. Only *one side* of the paper is to be used for printing. Printing should be consistently clear and dark.

Spacing. The text of the dissertation should be *double spaced*. Long quotations, footnotes, appendices, and references may be single spaced.

Color. It is recommended that you ***not*** rely on color to convey information (e.g., use symbols or labels rather than colors to differentiate the lines on a graph, or use stripes and cross-hatching instead of colors to distinguish areas on a map).

Photographs and Graphics. Photographs and graphics in the dissertation should be printed or photocopied directly on the paper as high quality black and white images. Scanned images must print clearly. If color must be used, ***only*** color laser or color photocopy printing is acceptable.

Use of materials copyrighted by others. IMPORTANT: Any material included that goes beyond "fair use" requires written permission of the copyright owner. It may be useful to include these in the dissertation as an appendix.

Pagination. Preliminary pages (i.e., the approval page, acknowledgments, table of contents, and the like) are to be numbered consecutively using lower case *Roman numerals*. All pages of the text, appendices (if any), and references must be numbered consecutively using *Arabic* numerals. The page number of the first page of each Chapter must be centered on the bottom of the page. In all other pages, the page number must be placed on the top of the page, aligned to the right.

Landscape pages. The top of a landscape page should be at the left margin and the bottom at the right margin. The page number is to be in the same relative position as on the portrait pages. An easy way to apply page numbers to landscape pages is to run them through the printer twice – once for the text, table, or figure (landscape orientation) and once for the page number (portrait orientation).

Sequence of the main components of the dissertation. The appropriate order of the major sections of the dissertation follows: the abstract, the title page, the copyright page (if needed), the approval page, acknowledgments, table of contents, the text, the bibliography/references, and appendices (if any).

Bibliography/References. The ACM (http://www.acm.org/pubs) or the IEEE (http://standards.ieee.org/resources) reference style should be followed.

Footnotes and Endnotes No specific rules. The format that is prescribed by your advisory committee should be followed.

* * *

It is recommended that you use your full legal name on the abstract, the title page, the copyright page (if appropriate), and on the approval page. Make certain that your name appears exactly the same way in all places.

Take a moment to check that all pages in all copies of your dissertation are accounted for and in the proper order before submitting final copies to the Departmental Secretary. *Two hardcopies* and *an electronic version* (in pdf format) of the dissertation should be submitted to the Departmental Secretary.

* * *

For students that plan to write their disseretation using Latex or MS Word authoring programms, they can download samples from http://www.cs.ucy.ac.cy/~chryssis/phd-specs.html.