



ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

Κωδικός	Τίτλος Μαθήματος	ΠΜ
ΕΠΑ131	Αρχές Προγραμματισμού	7.5

Διδάσκοντες Ελπίδα Κερανού-Παπαηλιού (email: elpida@cs.ucy.ac.cy) – τηλ: 22892694 –
Γραφείο: ΘΕΕ01, 117
Παύλος Αντωνίου (email: paul.antoniou@cs.ucy.ac.cy) – τηλ: 22893927 –
Γραφείο: ΘΕΕ01, B109

Εξάμηνο Εαρινό **Ακαδημαϊκό Έτος** 2017/18

Ημέρα/Ωρα διδασκαλίας	Ημέρα	Διάλεξη	Εργαστήριο	Φροντιστήριο
	Τρίτη, Παρασκευή	09:00-10:29, ΘΕΕ01, 202		
	Τετάρτη			10:00-10:59, ΣΘΕΕ01, 202
	Τρίτη, Παρασκευή		14:00-15:59 ΘΕΕ01, 103	

Ωρες γραφείου	Ελπίδα Κερανού-Παπαηλιού	Τετάρτη 10:00 – 12:00 ή κατόπιν συνεννοήσεως
----------------------	--------------------------	--

Επίπεδο μαθήματος	Προπτυχιακό	Βασικό	X
		Ενδιάμεσο	<input type="checkbox"/>
		Προχωρημένο	<input type="checkbox"/>
		Εξειδικευμένο	<input type="checkbox"/>
	Μεταπτυχιακό	<input type="checkbox"/>	

Κατηγορία μαθήματος	Περιγραφή	Προσφορές
Κορμού	Προγράμματα σπουδών στα οποία το μάθημα εντάσσεται ως υποχρεωτικό μάθημα.	<ul style="list-style-type: none">• Πληροφορικής• Δευτερεύον πρόγραμμα Πληροφορικής
Υποστηρίξης	Προγράμματα σπουδών στα οποία το μάθημα εντάσσεται ως μάθημα περιορισμένης επιλογής.	<input type="checkbox"/>
Δευτερεύον (Μόνο για Προπτυχιακά μαθήματα)	Προσφέρεται αποκλειστικά ως μάθημα ελεύθερης επιλογής.	<input type="checkbox"/>
	Προσφέρεται και ως μάθημα ελεύθερης επιλογής.	X
	Δεν προσφέρεται ως μάθημα ελεύθερης επιλογής.	<input type="checkbox"/>

Σκοπός Η ανάπτυξη δεξιοτήτων στην επίλυση προβλημάτων, με αλγοριθμικό τρόπο, και η θεμελίωση της αλγοριθμικής σκέψης, τόσο σε επίπεδο αρχών προγραμματισμού, όσο και σε επίπεδο μίας υψηλής επιπέδου γλώσσας προγραμματισμού (Java).

Αναμενόμενα Ο/η φοιτητής/φοιτήτρια που θα έχει ολοκληρώσει επιτυχώς το μάθημα αυτό,

**μαθησιακά
αποτελέσματα
και δεξιότητες
που θα
αναπτυχθούν**

αναμένεται ότι θα είναι σε θέση να:

1. Επεξηγεί τις εξής θεμελιώδεις έννοιες του δομημένου προγραμματισμού (structured programming): τιμές, τύποι (δεδομένων), μεταβλητές, παράμετροι, ορίσματα, συναρτήσεις/μεθόδους, εμβέλεις, δομή και διάρθρωση προγραμμάτων, βιβλιοθήκες και πελάτες, αρθρωτά προγράμματα, αφαιρετικότητα προγράμματος/δεδομένων, απόκρυψη πληροφοριών, επικοινωνία-διεπαφή.
2. Επεξηγεί τις εξής βασικές αλγοριθμικές δομές που μπορούν να χρησιμοποιηθούν στην «κατασκευή» προγραμμάτων: ακολουθία-διαδοχή, επανάληψη-αναδρομικότητα, επιλογή, μέθοδοι/συναρτήσεις.
3. Διακρίνει τις βασικές: (α) αρχές προγραμματισμού, (β) αλγοριθμικές τεχνικές και (γ) δομές προγραμμάτων.
4. Αναγνωρίζει τις θεμελιώδεις αρχές της επαναχρησιμοποίησης, άρθρωσης και ιεραρχικής δόμησης.
5. Επιδεικνύει δεξιότητες στην επίλυση προβλημάτων με χρήση του διαδικασιακού και αντικειμενοστραφούς μοντέλου προγραμματισμού.
6. Σχεδιάζει, υλοποιεί, δοκιμάζει (με οργανωμένο και συστηματικό τρόπο), αποσφαλματώνει και τεκμηριώνει αρθρωτά (modular) προγράμματα (μικρού μεγέθους), εφαρμόζοντας τη μεθοδολογία της σταδιακής ή προοδευτικής εκλέπτυνσης (stepwise refinement).
7. Επιχειρηματολογεί αναφορικά με την ορθότητα και εν γένει συμπεριφορά ενός προγράμματος και αξιολογεί βάσει ποιοτικών και άλλων κριτηρίων εναλλακτικές (αλγοριθμικές) λύσεις για το ίδιο πρόβλημα.
8. Αξιολογεί τη δυνατότητα επέκτασης (scaling up) ενός προγράμματος με στόχο τη διεύρυνση των προβλημάτων που μπορεί να καλύψει.
9. Επιδεικνύει ικανότητα αλγοριθμικής σκέψης, ανεξαρτήτως οποιασδήποτε γλώσσας προγραμματισμού
10. Επιδεικνύει ικανότητα στη χρήση της υψηλού επιπέδου γλώσσας προγραμματισμού, Java.

**Αναλυτική
περιγραφή**

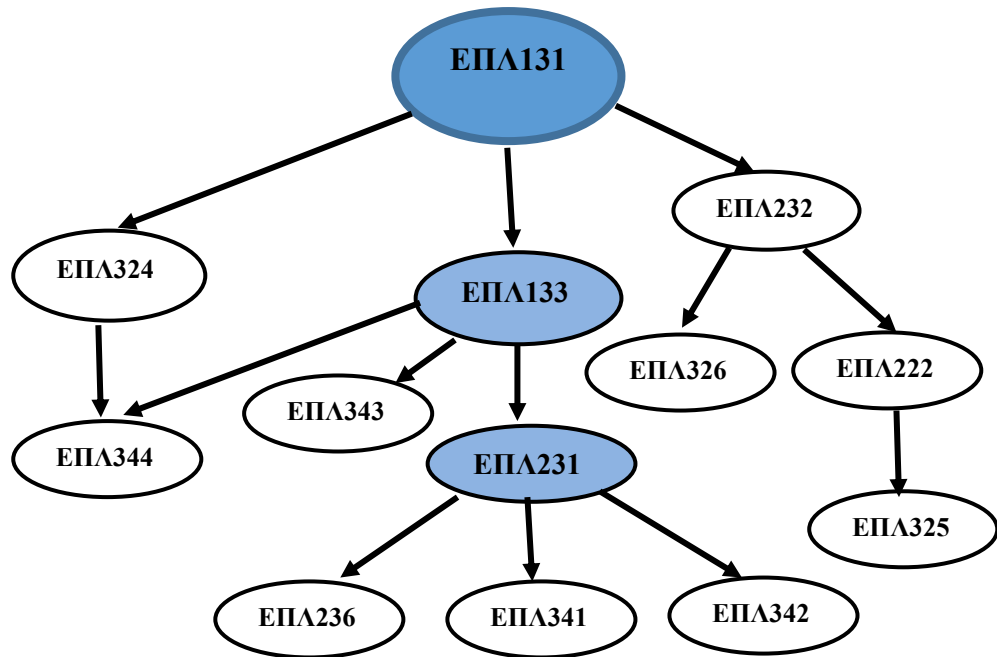
Εισαγωγή : Σημασία Βασικών Αρχών Προγραμματισμού

Ο προγραμματισμός αποτελεί αναπόσπαστο μέρος της Πληροφορικής. Το μάθημα ΕΠΛ131 σε Αρχές Προγραμματισμού είναι το πρώτο μάθημα της ευρύτερης περιοχής «Επίλυση Προβλημάτων». Στο μάθημα διδάσκονται οι βασικές, υψηλού επιπέδου αρχές προγραμματισμού κάνοντας χρήση της γλώσσας Java.

Πέραν των θεμελιωδών αρχών προγραμματισμού (δόμηση, άρθρωση, αφαιρετικότητα, κτλ), η διδασκαλία των οποίων αποτελεί τον κεντρικό στόχο του μαθήματος, το μάθημα καλύπτει πολύ βασικές έννοιες αναφορικά με την αρχιτεκτονική και λειτουργία των ηλεκτρονικών υπολογιστών.

Οι σχέσεις αυτού του μαθήματος με άλλα υποχρεωτικά μαθήματα του προγράμματος σπουδών στην Πληροφορική όλων των ειδικοτήσεων, δίνονται στο σχεδιάγραμμα που ακολουθεί. Είναι σημαντικό να γίνουν αντιληπτές αυτές οι

σχέσεις, ότι δηλαδή το ΕΠΑ131 αποτελεί αρχικό κόμβο σε οκτώ αλυσίδες υποχρεωτικών μαθημάτων. Και τούτο επειδή καλή γνώση των κεντρικών αρχών προγραμματισμού και εν γένει η θεμελίωση της αλγοριθμικής σκέψης, αποτελούν βασική προϋπόθεση για την περαιτέρω εξέλιξη της πορείας σπουδών στην Πληροφορική.



→ προαπαιτείται-από

- ΕΠΑ131 Αρχές Προγραμματισμού (1° Εξάμηνο)
- ΕΠΑ133 Αντικειμενοστραφής Προγραμματισμός (2° Εξάμηνο)
- ΕΠΑ231 Δομές Δεδομένων και Αλγόριθμοι (3° Εξάμηνο)
- ΕΠΑ232 Προγραμματιστικές Τεχνικές και Εργαλεία (3° Εξάμηνο)
- ΕΠΑ222 Λειτουργικά Συστήματα (4° Εξάμηνο)
- ΕΠΑ236 Αλγόριθμοι και Πολυπλοκότητα (4° Εξάμηνο)
- ΕΠΑ324 Επικοινωνίες και Δίκτυα (5° Εξάμηνο)
- ΕΠΑ341 Τεχνητή Νοημοσύνη (5° Εξάμηνο)
- ΕΠΑ342 Βάσεις Δεδομένων (5° Εξάμηνο)
- ΕΠΑ343 Τεχνολογία Λογισμικού (5° Εξάμηνο)
- ΕΠΑ344 Τεχνολογίες Διαδικτύου (6° Εξάμηνο)
- ΕΠΑ325 Παράλληλη Επεξεργασία (6° Εξάμηνο)
- ΕΠΑ326 Ασφάλεια Συστημάτων (6° Εξάμηνο)

Οι θεμελιώδεις αρχές στην επίλυση προβλημάτων και στον προγραμματισμό που διδάσκονται στο ΕΠΑ131 μέσω του διαδικασιακού και αντικειμενοστραφούς μοντέλου προγραμματισμού, ολοκληρώνονται στο μάθημα ΕΠΑ133 και πάλι με χρήση της γλώσσας Java, όπου καλύπτονται προχωρημένες αρχές και έννοιες του αντικειμενοστραφούς προγραμματισμού και εφαρμόζεται η αντικειμενοστραφής μεθοδολογία για την επίλυση πολύπλοκων προβλημάτων. Το μάθημα ΕΠΑ133 προαπαιτείται από το μάθημα ΕΠΑ231 όπου και πάλι με τη χρήση της γλώσσας Java διδάσκονται οι δομές δεδομένων και οι αλγόριθμοι επεξεργασίας τους καθώς και οι τεχνικές ανάλυσης της αποδοτικότητας αλγορίθμων. Όπως διαφαίνεται και από το πιο πάνω σχεδιάγραμμα η αλυσίδα μαθημάτων ΕΠΑ131 – ΕΠΑ133 – ΕΠΑ231 συνιστά κεντρική αλυσίδα του προγράμματος σπουδών, αφού από αυτήν διακλαδώνουν οκτώ διαδρομές που εμπλέκουν δέκα άλλα υποχρεωτικά μαθήματα από τις διάφορες περιοχές και ειδিকেύσεις του προγράμματος σπουδών. Από αυτά τα μαθήματα, άμεση σχέση με τον προγραμματισμό έχει επίσης το μάθημα ΕΠΑ232 το

οποίο καλύπτει προχωρημένες προγραμματιστικές τεχνικές και εργαλεία με τη χρήση μιας γλώσσας προγραμματισμού, η οποία μεταγλωττίζεται σε κώδικα μηχανής, π.χ. τη γλώσσα C. Στο πλαίσιο του μαθήματος αυτού καλύπτονται και βασικές εντολές του λειτουργικού συστήματος UNIX για προγραμματιστές.

Αναλυτική Περιγραφή Περιεχομένου

1. Εισαγωγή στους υπολογιστές και τις γλώσσες προγραμματισμού.
2. Επίλυση προβλημάτων και προγραμματισμός, προδιαγραφή προβλήματος, η έννοια του αλγόριθμου και του προγράμματος, μεθοδολογία προοδευτική εκτέλεση (stepwise refinement), υψηλού επιπέδου γλώσσες προγραμματισμού (σύνταξη και σημασιολογία), αφαιρετικότητα προγράμματος και δεδομένων, δόμηση, άρθρωση, κλάσεις και μέθοδοι (συναρτήσεις), εκτίμηση επίδοσης, τεκμηρίωση, γιατί Java.
3. Προγραμματισμός ως καταρτισμός σχεδίων δράσης, σχεδιασμός μεθόδων, ιεραρχική δομή, αλγοριθμικές δομές (ακολουθία/διαδοχή, επανάληψη, επιλογή, μέθοδοι/αφααιρετικότητα, μνήμη), φώλιασμα αλγοριθμικών δομών, βασικές κατηγορίες μεθόδων στη Java (στατικές μέθοδοι/συναρτήσεις, μέθοδοι αναφοράς, αφαιρετικότητα δεδομένων), η γενική αρχή του «διαίρει και βασίλευε».
4. Διεργασία ανάπτυξης λογισμικού, σχεδίαση από πάνω προς τα κάτω, διάσπαση προβλήματος, επαναχρησιμοποίηση, απόκρυψη πληροφοριών (διαχωρισμός ανάμεσα στο 'τί' και το 'πώς'), στρατηγικές δοκιμής και αποσφαλμάτωσης.
5. Ενσωματωμένοι τύποι δεδομένων, βασικοί ή βαθμωτοί τύποι (χαρακτήρες, αριθμοί, λογικές τιμές), συμβολοσειρές (τύπος String), μεταβλητές (ονομασίες, τύποι, τιμές, διευθύνσεις, εντολή ανάθεσης τιμών), τελεστές και εκφράσεις, κανόνες αποτίμησης εκφράσεων, μετατροπή τύπων δεδομένων (αυτόματα, ρητή μετατροπή – casting), ανακρίβειες αναπαράστασης.
6. Εμβάθυνση στις αλγοριθμικές δομές της επιλογής και επανάληψης: συνθήκες, λογικές εκφράσεις, βρόγχοι, κανόνες De Morgan, φώλιασμα επιλογών, διακλάδωση μόνο στην κατεύθυνση του όχι, επιλογές με πλήρη κάλυψη ενδεχομένων, ασαφείς επιλογές, εντολές if, if-else, switch, while, for, do-while.
7. Δομημένοι τύποι δεδομένων, μονοδιάστατοι πίνακες (ορισμός, δημιουργία – τελεστής new, αρχικοποίηση), αναπαράσταση πινάκων στη μνήμη, επεξεργασία πινάκων, πίνακες πολλαπλών διαστάσεων.
8. Είσοδος και έξοδος: είσοδος από γραμμή εντολής, standard είσοδος (πληκτρολόγιο) και έξοδος (οθόνη), επανα-κατεύθυνση εισόδου/εξόδου σε αρχεία (file redirection), σωλήνωση διεργασιών σε επίπεδο λειτουργικού συστήματος (process piping).
9. Συναρτήσεις (στατικές μέθοδοι) ως τα κατασκευαστικά τεμάχια για τη δημιουργία αρθρωτών προγραμμάτων, διεπαφή και ανατομία συναρτήσεων, παράμετροι, κλήσεις συναρτήσεων και ροή ελέγχου, ορίσματα, πέρασμα δια τιμής ή μέσω διευθύνσεως, κανόνες εμβέλειας ονομασιών, υπερφόρτωση συναρτήσεων (function overloading),
10. Βιβλιοθήκες και πελάτες (libraries and clients), συμβόλαιο μεταξύ πελάτη και υλοποίησης (API – Application Program Interface), η βιβλιοθήκη/κλάση String.

11. Αναδρομή (η δίδυμη αδελφή της επανάληψης), αναδρομή ουράς και στοίβας (tail and stack recursion), η χρήση της αρχής του «διαίρει και βασίλευε», ζητήματα επίδοσης.
12. Αντικείμενα (objects), κλάσεις αντικειμένων και στιγμιότυπα (instances), κατασκευαστές και μέθοδοι αναφοράς (reference methods), τελεστής new για κατανομή μνήμης, βασικοί τύποι έναντι τύπων αναφορών (reference types), αποθήκευση αντικειμένων, δημόσια/ιδιωτικά πεδία (public/private fields), στατικές μέθοδοι για την κλάση (static methods) και μη στατικές μέθοδοι (μέθοδοι αναφοράς για κάθε στιγμιότυπο αντικειμένου), συμβολοσειρές και πίνακες ως ενσωματωμένα αντικείμενα, διαχείριση μνήμης και σκουπιδιάρης (garbage collector) για επανάκτηση μνήμης, σχεδιάζοντας και δημιουργώντας τους δικούς μας αφηρημένους, νέους τύπους δεδομένων (αντικείμενα).

**Γλώσσα
διδασκαλίας**

Ελληνική

**Μέθοδοι
διδασκαλίας και
μάθησης**

Η διδασκαλία του μαθήματος αποτελείται από διαλέξεις, φροντιστήρια και πρακτική εξάσκηση, τόσο στο πλαίσιο οργανωμένων εργαστηρίων, όσο και κατ' ιδίαν εξάσκηση. **Η παρακολούθηση των διαλέξεων, φροντιστηρίων και εργαστηρίων είναι υποχρεωτική.**

Η καλλιέργεια και εμπέδωση της αλγοριθμικής σκέψης, καθώς και η ανάπτυξη δεξιοτήτων στον προγραμματισμό, απαιτούν εκτενή εξάσκηση στην επίλυση προβλημάτων με αλγοριθμικό τρόπο. Τα πρώτα προβλήματα που εξετάζονται είναι μικρά και απλά, και σταδιακά επεκτείνονται σε μέγεθος και πολυπλοκότητα. Η εξάσκηση μέσω προγραμματιστικών ασκήσεων είναι πολύμορφη. Κάποιες ασκήσεις είναι θεωρητικές και αφορούν (α) την ανάπτυξη αλγορίθμων στο χαρτί, σε επίπεδο ψευδοκώδικα, για την επίλυση δεδομένων προβλημάτων, ή (β) την «απόδειξη» της ορθότητας ή κάποιας ιδιότητας ενός αλγορίθμου, ή την κριτική ανάλυση της συμπεριφοράς ενός αλγορίθμου. Άλλες ασκήσεις συνεπάγονται, πέραν του σχεδιασμού των σχετικών αλγορίθμων, της υλοποίησής τους σε μορφή κώδικα στη γλώσσα Java, και της εκτέλεσης, δοκιμής και αποσφαλμάτωσης των προγραμμάτων αυτών. Σε κάθε περίπτωση, είτε η λύση ενός προβλήματος διατυπώνεται σε ψευδογλώσσα, είτε σε εκτελέσιμο κώδικα, η παρουσίαση της λύσης και η τεκμηρίωσή της ώστε να είναι αυτό-επεξηγηματική αποτελεί σημαντικό κριτήριο για την ποιότητα της λύσης.

Οι υποχρεωτικές ασκήσεις για την απαιτούμενη πρακτική εξάσκηση είναι δύο μορφών, προγραμματιστικές ασκήσεις και διαγνωστικά, όπως εξηγείται στην επόμενη ενότητα. Πέραν, όμως από τις υποχρεωτικές ασκήσεις, κάθε φοιτητής/φοιτήτρια αναμένεται ότι, στο πλαίσιο της αυτόνομης μελέτης του/της για το μάθημα αυτό, θα αντλήσει και από μόνος/η του/της επιπρόσθετες ασκήσεις για περαιτέρω εξάσκηση, καθαρά με δική του/της πρωτοβουλία. Οι πηγές από τις οποίες μπορεί ένας να αντλήσει επιπρόσθετες ασκήσεις είναι πολλές, για παράδειγμα: (α) από τα βιβλία που στηρίζουν το μάθημα, (β) από τα προβλήματα που αναφέρονται στις διαλέξεις και τα φροντιστήρια, ή (γ) από προβλήματα της καθημερινότητας δικής του/της επινόησης/ επιλογής. Στον προγραμματισμό, η πρακτική τριβή με όσο γίνεται περισσότερα προβλήματα, είναι εξίσου αν όχι πιο σημαντική από την καθαρά θεωρητική μελέτη των θεμάτων. Επισημαίνεται, ιδιαίτερα, ότι πρακτική τριβή δε σημαίνει απλά τη στείρα παραγωγή κώδικα (μεταφράζοντας ενδεχομένως δεδομένους αλγορίθμους), αλλά πρωτίστως το σχεδιασμό αλγορίθμων για την επίλυση προβλημάτων. Επί της ουσίας, ο σχεδιασμός αλγορίθμων είναι το δημιουργικό στοιχείο του προγραμματισμού.

Τα οργανωμένα εργαστήρια στοχεύουν στην εξοικείωση των διδασκομένων με τα υπολογιστικά συστήματα που στηρίζουν το μάθημα (όπου παρουσιάζονται τα σχετικά εργαλεία), καθώς επίσης στην περαιτέρω εμπέδωση, σε καθαρά πρακτικό επίπεδο, των εννοιών που καλύπτονται στις διαλέξεις, χρησιμοποιώντας προβλήματα/ασκήσεις, αλγόριθμους και προγράμματα που εξετάζονται στις διαλέξεις και φροντιστήρια. Στα εργαστήρια θα διεξάγονται και τα διαγνωστικά tests για ανατροφοδότηση των διδασκομένων σε σχέση με τους μαθησιακούς στόχους του μαθήματος και ως εκ τούτου ως μέσου ενίσχυσης της αυτογνωσίας των διδασκομένων. Μέρος του χρόνου που διατίθεται για την υποχρεωτική εργαστηριακή εξάσκηση μπορεί να αξιοποιηθεί για την εκπόνηση υποχρεωτικών ασκήσεων.

Συνεπώς, οι μέθοδοι διδασκαλίας και μάθησης είναι οι εξής:

(α) **Διαλέξεις**, όπου παρουσιάζονται και εξετάζονται με κριτική διάθεση οι διάφορες έννοιες που καλύπτει το μάθημα. Η παρακολούθηση των διαλέξεων είναι υποχρεωτική. Οι φοιτητές/φοιτήτριες ενθαρρύνονται να συμμετέχουν στις συζητήσεις που γίνονται. Με άλλα λόγια επιδιώκεται ο διάλογος και η ανταλλαγή επιχειρημάτων σε σχέση με τα θέματα που παρουσιάζονται.

(β) **Φροντιστήρια**, κατά τα οποία δεν παρουσιάζονται νέες έννοιες αλλά στόχος είναι η περαιτέρω εμπέδωση των εννοιών που καλύπτονται στις διαλέξεις και ο εμπλουτισμός της πραγμάτευσης του όλου θέματος των αρχών προγραμματισμού με επιπρόσθετες ασκήσεις και συζητήσεις. Η παρακολούθηση των φροντιστηρίων είναι υποχρεωτική. Επιπρόσθετα, στο πλαίσιο των φροντιστηρίων το διαλογικό στοιχείο ενισχύεται και αναμένεται να είναι το κυρίαρχο στοιχείο των συναντήσεων, όπου μάλιστα σε μεγάλο βαθμό η πραγμάτευση καθοδηγείται από τους φοιτητές/φοιτήτριες και όχι τους διδάσκοντες. Με άλλα λόγια οι φοιτητές/φοιτήτριες ενθαρρύνονται να εισηγούνται θέματα/ασκήσεις προς συζήτηση, για παράδειγμα θέματα που φαίνεται να δημιουργούν κάποιες δυσκολίες.

(γ) **Εργαστήρια**, για την παρουσίαση των σχετικών πληροφοριακών εργαλείων (που είτε στηρίζουν το μάθημα, είτε αποτελούν χρήσιμα εργαλεία για τον προγραμματισμό), την εμπέδωση σε καθαρά πρακτικό επίπεδο των εννοιών του προγραμματισμού, και γενικά τη στήριξη της πρακτικής εξάσκησης διαθέτοντας, για συγκεκριμένο χρόνο κάθε εβδομάδα, σχετικά εργαστήρια του Τμήματος Πληροφορικής. Η παρουσία στο χώρο του εργαστηρίου κατά τις εν λόγω περιόδους είναι υποχρεωτική. Τα διαγνωστικά tests διεξάγονται πρωτίστως κατά τη διάρκεια εργαστηρίων.

(δ) **Ασκήσεις και πρακτική εξάσκηση**, όπου σε μεγάλο βαθμό, ο/η κάθε φοιτητής/φοιτήτρια οργανώνει μόνος/η του/της το χρόνο του/της. Οι ασκήσεις συζητούνται στο πλαίσιο των διαλέξεων και εργαστηρίων. Όμως αυτό το στοιχείο του μαθήματος αφορά πρωτίστως την κατ' ιδίαν απασχόληση και προβληματισμό, σε σχέση με όλες τις πτυχές της επίλυσης προβλημάτων με αλγοριθμικό τρόπο, μέσω ασκήσεων (που είτε αποτελούν μέρος προγραμματισμένης υποχρεωτικής εξάσκησης ή έχουν επιλεγεί από τον/την φοιτητή/φοιτήτρια ως επιπρόσθετη προαιρετική εξάσκηση).

Η επιτυχία σε κάθε ένα από τα δέκα μαθησιακά αποτελέσματα που προσδοκούνται με την ολοκλήρωση του μαθήματος, συνεπάγεται την αξιοποίηση από μέρους των διδασκομένων και των τεσσάρων μεθόδων διδασκαλίας και μάθησης που αναφέρονται πιο πάνω. Με άλλα λόγια, για κάθε μαθησιακό στόχο απαιτούνται και οι τέσσερις μέθοδοι διδασκαλίας και μάθησης ως αναπόσπαστα στοιχεία μίας ενιαίας προσέγγισης για επιτυχή ολοκλήρωση.

Τρόποι αξιολόγησης

Η επίδοση των φοιτητών/φοιτητριών αξιολογείται συνεχώς με βάση δύο τύπους υποχρεωτικών εργασιών: (α) προγραμματιστικές ασκήσεις και (β) διαγνωστικά. Η εκπόνηση όλων των εργασιών είναι υποχρεωτική.

Προγραμματιστικές ασκήσεις

Κάθε προγραμματιστική άσκηση πρέπει να εκπονείται ατομικά, εκτός αν δοθούν διαφορετικές οδηγίες. Εργασίες που παραδίδονται μετά την προθεσμία, μπορεί να μη ληφθούν υπόψη. Η κάθε εργασία αποτελείται από:

(α) Σύντομη τεκμηρίωση της σχεδίασης του προγράμματος, στην οποία να δίνεται η διάσπαση του προβλήματος σε υποπροβλήματα, και η ανάλυση των αλγορίθμων των κυριότερων υποπροβλημάτων.

(β) Λίστα του κώδικα (σημειώνεται ότι σχόλια μέσα στον κώδικα αποτελούν αναγκαίο στοιχείο).

(γ) Λίστα της δοκιμής του κώδικα.

Κάθε πρόγραμμα υποβάλλεται σε ηλεκτρονική μορφή μέσω ηλεκτρονικού ταχυδρομείου ή του σχετικού εργαλείου τηλεμάθησης που στηρίζει το μάθημα. Επιπρόσθετα, το πρόγραμμα, η τεκμηρίωσή του, και το script των δοκιμών υποβάλλονται σε έντυπη μορφή.

Η ανταλλαγή απόψεων πάνω στις πρακτικές ασκήσεις συνιστάται. **Τονίζεται, όμως, ότι η αντιγραφή κώδικα, ή η παροχή κώδικα προς αντιγραφή, ή η τροποποίηση των αποτελεσμάτων της δοκιμής του κώδικα, αποτελούν πράξεις δόλου που στην καλύτερη περίπτωση οδηγούν σε μηδενισμό.**

Οι ασκήσεις διορθώνονται και επιστρέφονται εγκαίρως. Ο/η φοιτητής/ φοιτήτρια έχει την ευκαιρία να συζητήσει την εργασία του/της στα φροντιστήρια ή στις ώρες γραφείου των διδασκόντων. Οι λύσεις θα αναρτώνται στο διαδικτυακό τόπο του μαθήματος.

Συνιστάται στους/στις φοιτητές/φοιτήτριες να κρατούν τις προγραμματιστικές εργασίες τους μέχρι την ολοκλήρωση του μαθήματος.

Διαγνωστικά

Κάποια από τα εργαστήρια θα έχουν μορφή διαγνωστικών. Ένα διαγνωστικό είναι μια προγραμματιστική εργασία υπό τη μορφή εξέτασης που θα εκτελείται ατομικά σε εργαστήριο υπολογιστών υπό την επίβλεψη διδακτικού προσωπικού. Η ημερομηνία διεξαγωγής του κάθε διαγνωστικού θα ανακοινώνεται περίπου μια εβδομάδα πριν, προφορικά (στην τάξη) και ηλεκτρονικά (στην ιστοσελίδα του μαθήματος). Οι λύσεις θα αναρτώνται στο διαδικτυακό τόπο του μαθήματος. Μικρά διαγνωστικά πιθανόν να γίνουν και στη διάρκεια κάποιων εκ των φροντιστηρίων. Σε αυτήν την περίπτωση, ένα διαγνωστικό θα είναι μια ή περισσότερες προγραμματιστικές ασκήσεις που θα πρέπει να λυθούν στο χαρτί. Η ημερομηνία διεξαγωγής αυτών των διαγνωστικών θα ανακοινώνεται την προηγούμενη μέρα.

Βαθμολογία

Προγραμματιστικές ασκήσεις και Διαγνωστικά	30%
Ενδιάμεση γραπτή εξέταση	20%
Τελική γραπτή εξέταση	50%

Οι πιο πάνω μέθοδοι αξιολόγησης εφαρμόζονται με διαφάνεια και συνέπεια.

Τεκμηρίωση πιστωτικών μονάδων

Συγκεκριμένα, κάθε φοιτητής/φοιτήτρια λαμβάνει πίσω τις εργασίες του/της για κάθε υποχρεωτική άσκηση, όπως και για την ενδιάμεση εξέταση, μαζί με την ανάλυση της βαθμολογίας του/της καθώς και υποδειγματικές λύσεις/απαντήσεις. Επιπρόσθετα, οι λύσεις/απαντήσεις των φοιτητών/φοιτητριών μπορεί να συζητηθούν σε φροντιστήρια ή κατά τις ώρες γραφείου. Η τελική εξέταση, που έχει και τη μεγαλύτερη βαρύτητα, δεν επιστρέφεται, αλλά και πάλι ο/η φοιτητής/φοιτήτρια μπορεί να ζητήσει από τους διδάσκοντες να εξετάσουν μαζί τις απαντήσεις του/της και τις βαθμολογίες που πήρε για πλήρη και διαφανή τεκμηρίωση της επίδοσής του/της. Συλλογικά και αναπόσπαστα, οι πιο πάνω μέθοδοι αξιολόγησης (προγραμματιστικές ασκήσεις, διαγνωστικά και γραπτές εξετάσεις) συνιστούν επαρκείς και κατάλληλες μεθόδους για την έγκυρη πιστοποίηση της επιτυχούς (ή μη επιτυχούς) ικανοποίησης των δέκα προσδοκώμενων και αλληλένδετων μαθησιακών αποτελεσμάτων που υπογραμμίζουν την επιτυχή ολοκλήρωση του μαθήματος.

Κάθε πιστωτική μονάδα αντιστοιχεί σε 25-30 ώρες εργασίας από μέρους του μέσου φοιτητή/φοιτήτριας για την επιτυχή ολοκλήρωση των αναμενόμενων μαθησιακών αποτελεσμάτων. Συνεπώς, η επιτυχής ολοκλήρωση του μαθήματος συνεπάγεται 187.5-225 ώρες εργασίας. Αυτός είναι ο συνολικός φόρτος εργασίας για το μέσο φοιτητή/φοιτήτρια και αναλύεται ως εξής:

Παρακολούθηση Διαλέξεων και Φροντιστηρίων: 4 ώρες εβδομαδιαίως για 13 εβδομάδες, **σύνολο 52 ώρες.**

Κατ' ιδίαν διάβασμα και πρακτική εξάσκηση: 5 ώρες για 13 εβδομάδες, **σύνολο 65 ώρες** (σημειώνεται ότι στο χρόνο αυτό περιλαμβάνονται και δύο ώρες εβδομαδιαίως από τις εργαστηριακές συναντήσεις).

Εκπόνηση υποχρεωτικών ασκήσεων: **Σύνολο 52 ώρες** (σημειώνεται ότι στο χρόνο αυτό περιλαμβάνονται και δύο ώρες εβδομαδιαίως από τις εργαστηριακές συναντήσεις).

Γραπτές εξετάσεις: 8 ώρες προετοιμασία (κυρίως μέσω επίλυσης ασκήσεων στο χαρτί ή/και τον υπολογιστή) και 3 ώρες για την εξέταση, για κάθε μία από τις δύο γραπτές εξετάσεις, **σύνολο 22 ώρες.**

Συνεπώς, ο συνολικός φόρτος εργασίας για την επιτυχή ολοκλήρωση του μαθήματος από το μέσο φοιτητή/φοιτήτρια εκτιμάται στις **191 ώρες εργασίας.**

Βιβλιογραφία

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

Ιστοσελίδα μαθήματος: www.cs.ucy.ac.cy/courses/EPL131