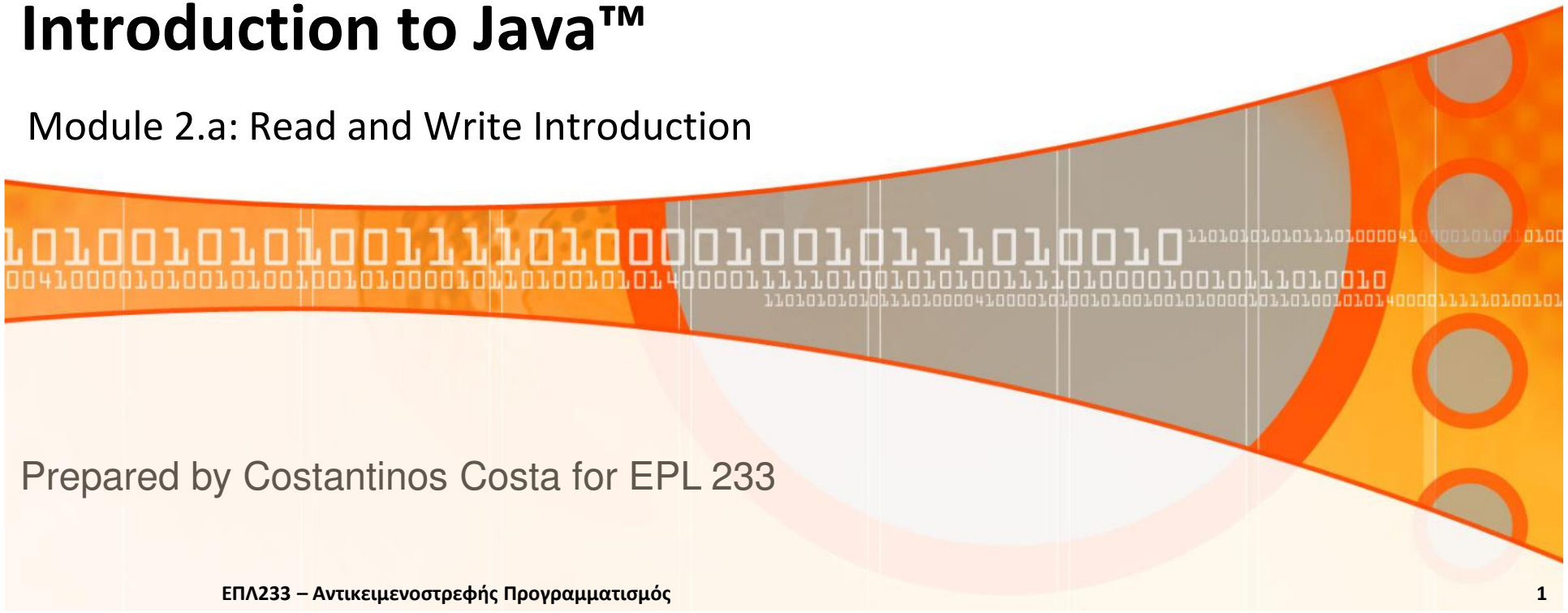
Introduction to Java™

Module 2.a: Read and Write Introduction



Prepared by Costantinos Costa for EPL 233

Java Input Review

- **CONSOLE:**
  - `Scanner stdin = new Scanner( System.in );`

  The "standard" input stream. This stream is already open and ready to supply input data. Typically this stream corresponds to keyboard input or another input source specified by the host environment or user.

- **FILE:**
  - `Scanner inFile =`

```
new Scanner( new FileReader(srcFileName ));
```

Java Output Review

- **CONSOLE:**
  - `System.out.print("To the screen");`

The "standard" output stream. This stream is already open and ready to accept output data. Typically this stream corresponds to display output or another output destination specified by the host environment or user.

- **FILE:**
  - `PrintWriter fout = new PrintWriter(new File("output.txt");`
  - `fout.print("To a file");`

Java Output Review

- **CONSOLE:**
  - `System.err.print(“Error to the screen");`

The "standard" error output stream. This stream is already open and ready to accept output data. Typically this stream corresponds to display output or another output destination specified by the host environment or user. By convention, this output stream is used to display error messages or other information that should come to the immediate attention of a user even if the principal output stream, the value of the variable out, has been redirected to a file or other destination that is typically not continuously monitored.

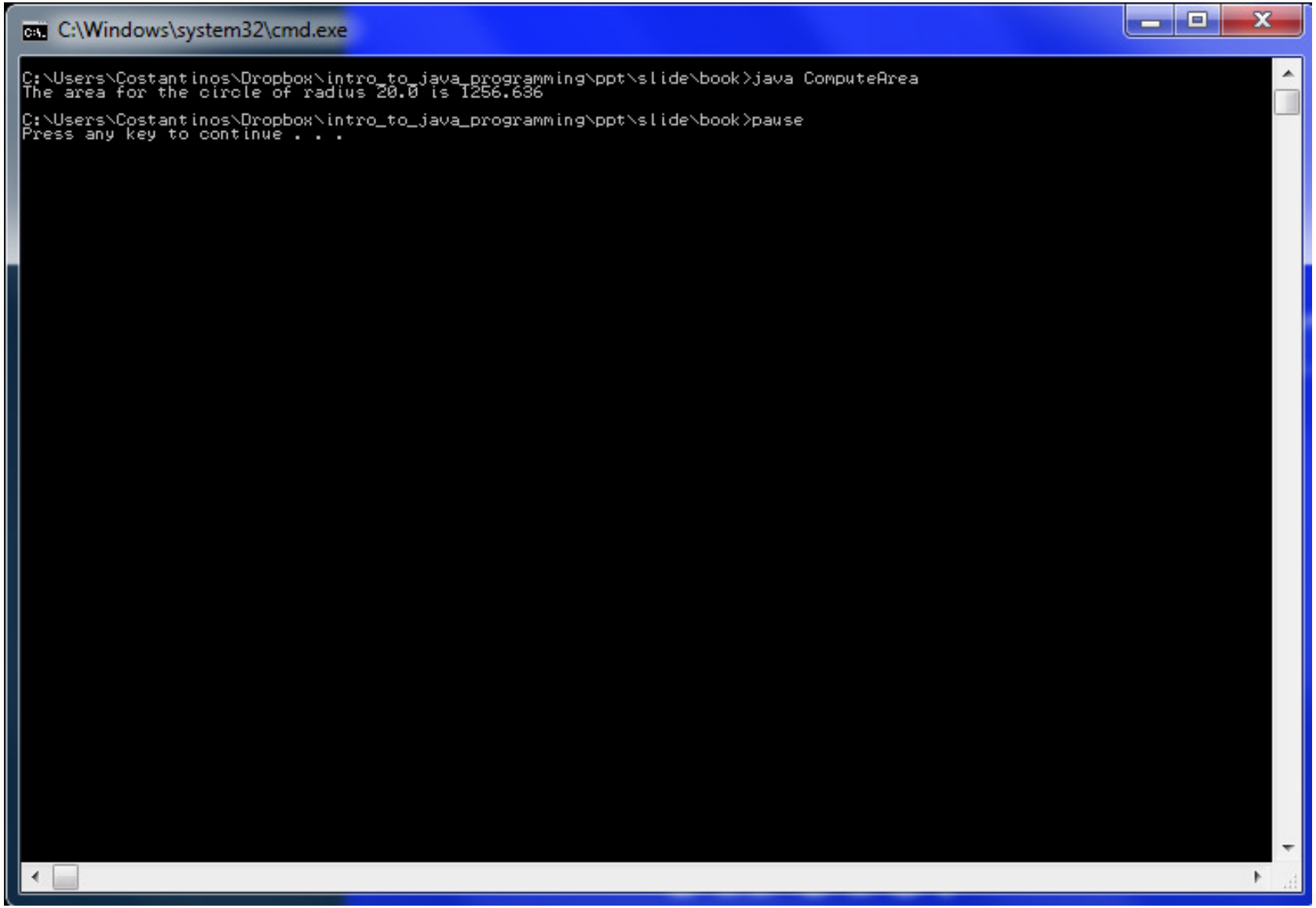
- **FILE:**
  - `PrintWriter fout = new PrintWriter(new File("output.txt");`
  - `fout.print("To a file");`

Example

```
public class ComputeArea {
 public static void main(String[] args) {
          double radius; // Declare radius
          double area; // Declare area

          // Assign a radius
          radius = 20; // New value is radius
          // Compute area
          area = radius * radius * 3.14159;
          // Display results
          System.out.println("The area for the circle of radius " + radius + " is " +
area);
          }
}
```

Output

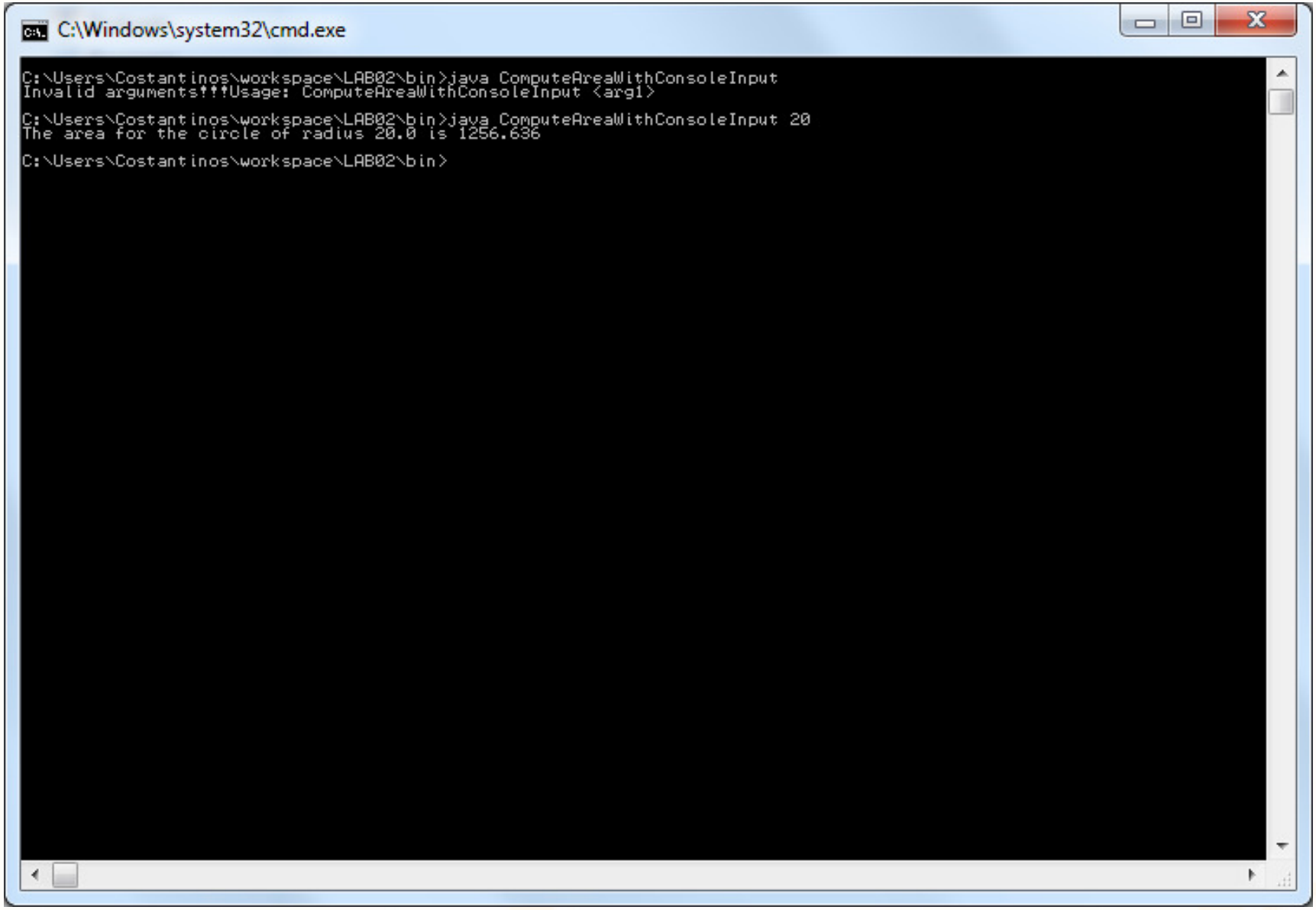

C:\Windows\system32\cmd.exe
C:\Users\Costantinos\Dropbox\intro_to_java_programming\ppt\slide\book>java ComputeArea
The area for the circle of radius 20.0 is 1256.636
C:\Users\Costantinos\Dropbox\intro_to_java_programming\ppt\slide\book>pause
Press any key to continue . . .

Reading Input from arguments

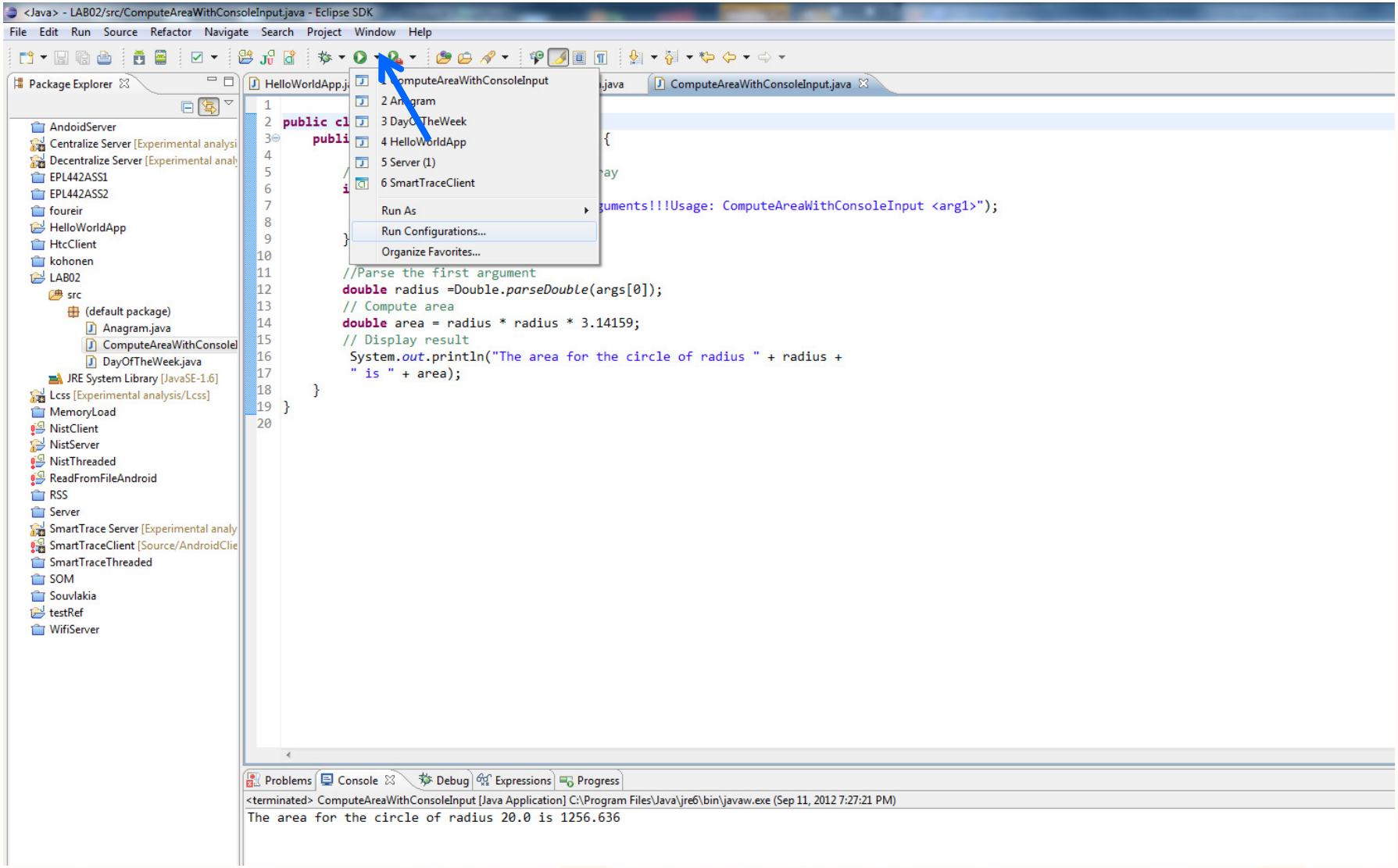
```
public class ComputeAreaWithConsoleInput {
    public static void main(String[] args) {

    //Check the length of the input array
    if (args.length < 1) {
    System.err.println("Invalid arguments!!!Usage: ComputeAreaWithConsoleInput
    <arg1>");
    return;
    }
    //Parse the first argument
    double radius =Double.parseDouble(args[0]);
    // Compute area
    double area = radius * radius * 3.14159;
    // Display result
     System.out.println("The area for the circle of radius " + radius +
     " is " + area);
    }
}
```

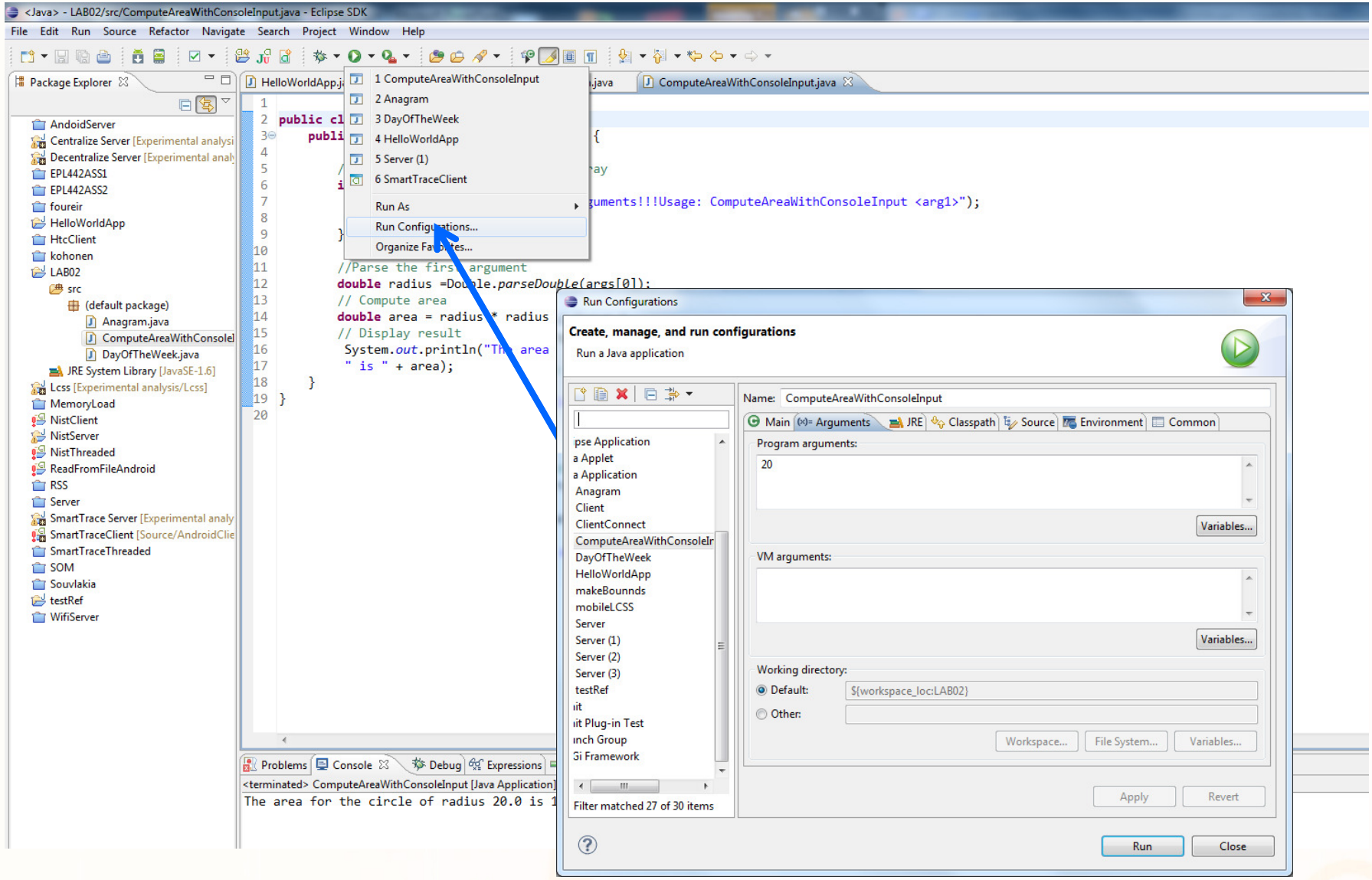
Output


C:\Windows\system32\cmd.exe
C:\Users\Costantinos\workspace\LAB02\bin>java ComputeAreaWithConsoleInput
Invalid arguments!!!Usage: ComputeAreaWithConsoleInput <arg1>
C:\Users\Costantinos\workspace\LAB02\bin>java ComputeAreaWithConsoleInput 20
The area for the circle of radius 20.0 is 1256.636
C:\Users\Costantinos\workspace\LAB02\bin>

Reading Input from arguments



Reading Input from arguments



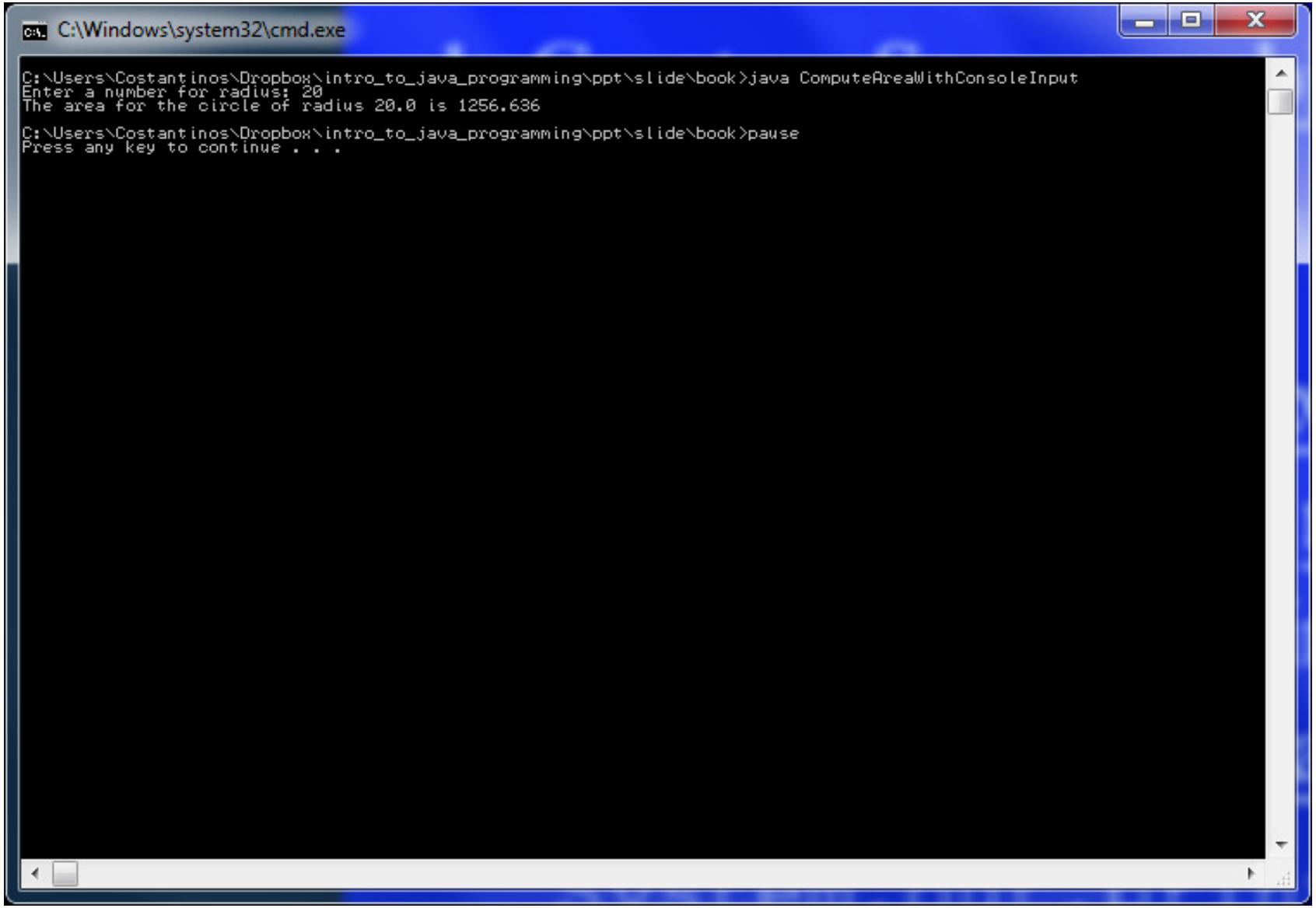
Reading Input from the Console

1. Create a Scanner object
   - `Scanner input = new Scanner(System.in);`
2. Use the methods next(), nextByte(), nextShort(), nextInt(), nextLong(), nextFloat(), nextDouble(), or nextBoolean() to obtain to a string, byte, short, int, long, float, double, or boolean value. For example,
   - `System.out.print("Enter a double value: ");`
   - `Scanner input = new Scanner(System.in);`
   - `double d = input.nextDouble();`

Reading Input from the Console

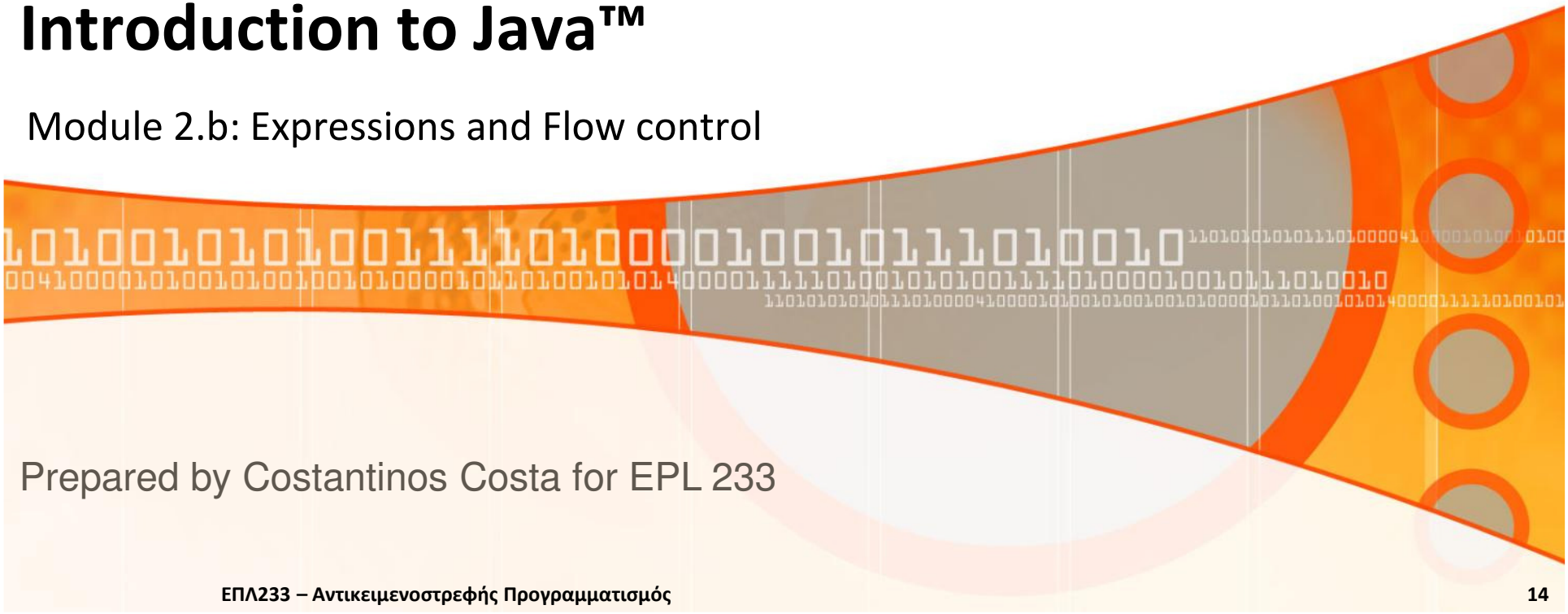
```
import java.util.Scanner; // Scanner is in the java.util package

public class ComputeAreaWithConsoleInput {
    public static void main(String[] args) {
        // Create a Scanner object
        Scanner input = new Scanner(System.in);
        // Prompt the user to enter a radius
        System.out.print("Enter a number for radius: ");
        double radius = input.nextDouble();
        // Compute area
        double area = radius * radius * 3.14159;
        // Display result System.out.println("The area for the circle of radius " + radius
        + " is " + area);
    }
}
```


C:\Windows\system32\cmd.exe
C:\Users\Costantinos\Dropbox\intro_to_java_programming\ppt\slide\book>java ComputeAreaWithConsoleInput
Enter a number for radius: 20
The area for the circle of radius 20.0 is 1256.636
C:\Users\Costantinos\Dropbox\intro_to_java_programming\ppt\slide\book>pause
Press any key to continue . . .

Introduction to Java™

Module 2.b: Expressions and Flow control



Prepared by Costantinos Costa for EPL 233

Operators

- Used as in C/C++
  - + - * / =
  - % (modulo)
  - ! (not)
  - || (or), && (and)
  - <,>, ==, >=, <=, <>,!=
  - n++, n--, ++n, --n

Casting

- variable= (type) variable;
  - Example:

```
void casts() {
  int I = 200;
  long l = (long) I;
  long l2 = (long)200;
}
```

- Getting an int out of a String:

```
Integer x=new Integer(String);
int xValue= x.intValue();
```

Similar for float, double, long etc…

Look it up in the API documentation.

Casts in fundamental types

- Casting from another type to int, float, short, long, double:
  - Create an instance of the appropriate wrapper class using the suitable constructor
  - Get the desired value using the appropriate property, usually the .[type]Value
- Example: (String to float)

```
String piV="3.14159";
Float temp=new Float(piV); //create a new Float object
float pi = temp.floatValue(); //get the float value of the Float
```

Casting examples

- Integer

```
String n = “23”;
Integer tmp = new Integer(n);
int i = tmp.intValue();
float f = tmp.floatValue();
double d = tmp.doubleValue();
Integer tmp2 = Integer.valueOf(“342”);
```

- Float

```
Float tmp = new Float(n);
int i = tmp.intValue();
float f = tmp.floatValue();
double d = tmp.doubleValue();
```

Flow control

- Java supports if-then-else, while, do-while, for and switch (case) statements as well as labeled breaks.
- The syntax for most of these statements is the same as in C/C++. (in addition variables can be declared in their conditions)
  - If-then-else

```
if (yoursales >=2*target)
 {bonus=1000;}
else if (yoursales >=1.5*target)
 {bonus=500;}
else if (yoursales >= target)
 {bonus=100;}
else {System.out.println("You are fired!!");}
```

Flow control

- while (just like C/C++)
  while (condiction) {block}

```
while(count<6)
 { System.out.println(count);
    count++;
 }
```

- do-while (just like C/C++)

```
do {block} while (condition)
```

Flow control

- for loops

```
for(statement1;condition;statement2)
 {statements;}

for(int x=0;x<100;x++)
  {System.out.println("Number is " + x);}
```

- switch statement

```
switch (choise) {
 case 1: {……;break;}
 case 2: {……;break;}
 default: {System.out.println("Invalid input");break;}
} //end switch
```

Flow control

- The break keyword
  - Used for breaking out of loops
    (and in the switch statement)

- The continue statement
  - Used for transferring the execution to the top of the loop

Flow control

- Labeled break/continue
  - Similar to GOTO
  - Convenient for breaking out of nested loops or transferring the execution at the top of the loop.
  - Label must be placed just outside the loop followed by :
  - Works with all kinds of loops

Flow control

- **Break/continue example**

```
outsideLoop: //label
  for(int out=0; out<3; out++) {
     //some code
   for(int inner=0;inner < 5; inner++) {
     //some code
     if (.....)
      {break outsideLoop;} //break out of the outer loop
  if (.....)
   {continue outsideLoop;} //continue to outer loop
  }//inner loop
 }//outer loop
 System.out.println("All done");
```

Example-Exercise

- Implement a Java-method that prints out the day of the week for a given day (1..31), month (1..12) and year. The day of the week of dates between March 1900 and February 2100 can be calculated as follows:
  - First, you have to calculate the total number of days from 1900/1/1 to the given date (see below, for details). Secondly, you divide this number by 7 with integer remainder: this now is the day of the week, with 0 as sunday, 1 as monday, etc.

- To calculate the total number of days you have to implement the following steps:
  - Subtract 1900 from the given year and multiply the result by 365
  - Add the missing leaps years by adding (year - 1900) / 4.
  - If the year itself is a leap year and the month is January or February, you have to subtract 1 from the previous result.
  - Now add the all days of the year up to the given one to the result (in case of february always 28, because the additional day for a leap year already have been added).
- Here some dates
  - Sunday: 4/4/2010, 9/9/2012
  - Wednesday: 17/2/2010, 12/9/2012

Solution

```
/**
 * For a given date the corresponding day of the week is printed out.
 *
 * @author Costantinos Costa
 *
 */
public class DayOfTheWeek {

      /**
       * Prints out the day of the week for a given day, month, and year.
       */
      public static void main(String[] args) {

      //Check the length of the input array
      if (args.length < 3) {
      System.err.println("Invalid arguments!!!Usage: DayOfTheWeek <arg1> <arg2> <arg3>");
      return;
      }

      int day = Integer.parseInt(args[0]);
      int month = Integer.parseInt(args[1]);
      int year = Integer.parseInt(args[2]);
```

Solution

```
// First we need to check that is a valid date
switch (month) {
      case 12:
      case 10:
      case 8:
      case 6:
      case 4:
            if (day > 31) {
            System.out.println("Invalid Date");
            return;}
      break;
      case 11:
      case 9:
      case 7:
      case 5:
      case 3:
      case 1:
            if (day > 31) {
            System.out.println("Invalid Date");
            Return;}
      break;
      case 2:
            if (day > 28) {
                  if (!(year % 4 == 0) || day != 29) {
                  System.out.println("Invalid Date");
                  return;}}
      break;
      default:
      System.out.println("Invalid Date");
      return;
}
```

Solution

```
int dayOfTheWeek = (year - 1900) * 365 + (year - 1900) / 4;
// In case that the year is a leap year and the month is smaller than
// February
if (year % 4 == 0 && month <= 2) {
dayOfTheWeek--;}

switch (month) {
      case 12:
      dayOfTheWeek += 30; // NO break!!! Fall through to all cases
      case 11:
      dayOfTheWeek += 31;
      case 10:
      dayOfTheWeek += 30;
      case 9:
      dayOfTheWeek += 31;
      case 8:
      dayOfTheWeek += 31;
      case 7:
      dayOfTheWeek += 30;
      case 6:
      dayOfTheWeek += 31;
      case 5:
      dayOfTheWeek += 30;
      case 4:
      dayOfTheWeek += 31;
      case 3:
      dayOfTheWeek += 28;
      case 2:
      dayOfTheWeek += 31;
}
```

Solution

```
        // Determine the day of the week by a simple modulo operation
        dayOfTheWeek = (day + dayOfTheWeek) % 7;

        switch (dayOfTheWeek) {
        case 0:
        System.out.println("Sunday");
        break;
        case 1:
        System.out.println("Monday");
        break;
        case 2:
        System.out.println("Tuesday");
        break;
        case 3:
        System.out.println("Wednesday");
        break;
        case 4:
        System.out.println("Thursday");
        break;
        case 5:
        System.out.println("Friday");
        break;
        case 6:
        System.out.println("Saturday");
        break;
        }
    }
}
```