

# Software Development

## Lab: Data types. Variables. Constants.

### Topics

1. Data types, variables and constants ([section 1](#))
2. Debug Java programs ([section 2](#))

Today we use the following tool/software:

- **TextPad**

- Use TextPad to write, compile and run your Java programs
- To launch the TextPad click on the ***Windows Start Menu*** → ***All Programs*** → ***TextPad*** → ***TextPad***
- Compiling and Running a Java program
  - \* **Step 1: Compile the program** – there are two options perform either option 1 or option 2
    1. From the TextPad's menu bar choose Tools → External Tools → Compile
    2. press at the same time the keys **Ctrl + 1**
  - \* **Step 2: Run the program** – there are two options perform either option 1 or option 2
    1. From the TextPad's menu bar choose Tools → External Tools → Run
    2. press at the same time the keys **Ctrl + 2**

## 1 Data types, variables and constants

### Problem 1

Write a java program that declares a variable to represent the side of a square. The side of the square is 7 units. Compute and display the area of the square. Save the program as Square.java

## Problem 2

Write a java class that declares variables to represent the length and the width of a room in metres and the price of carpet per metre in euros. Assign appropriate values to the variables. Compute and display the cost of carpeting the room. Save the class as Carpet.java.

## Problem 3

Convert the Carpet class to an interactive application. Instead of assigning values to the length, width and price variables accept them from the user as input. Save the revised class as InteractiveCarpet.java.

## Problem 4

We will write a program to perform arithmetic operations

1. Integer Multiplication
2. Integer Division, Double Division and Mixed Division
  - (a) **Double Division:** `double result = 10.0/4.0; //`  $\rightarrow 2.5$
  - (b) **Integer Division:** `int result = 10/4; //`  $\rightarrow 2$
  - (c) **Mixed Division:**
    - `int result = 10.0/4; //` compiling error: expects int but the result is of type double
    - `int result = (int) 10.0/4; //`  $\rightarrow 2$  We force the type we want by **type casting** to (**data type**), in this example (int)
    - Above, at item (b) the integer division is not the result we know from maths, if we want the correct division result we use type casting, in this example (double):
      - `double result = (double) 10/4 //`  $\rightarrow 2.5$
      - or
      - `double result = 10/(double)4 //`  $\rightarrow 2.5$
    - Note: make sure that you type cast one of the operands and not the result of the division
      - `double result = (double) (10/4); //`  $\rightarrow 2$  integer division

### Problem 5

**ChangeCalculator** is an application that helps a seller to give the right change to a customer. The application prompts the user to enter the total price of the items bought and the amount received from the customer. The application calculates and displays the change the customer is owed. In addition, the application calculates and displays the maximum number of how many EUR 50, EUR 20, EUR 10 and EUR 5 notes, and how many EUR 2 and EUR 1 coins the seller has to give to the customer as change. For simplicity, let's consider that the seller does not want to deal with small coins (i.e. 50c, 20c, etc.), therefore although the total price of the items bought may be a decimal number the seller will enter the price rounded down to the nearest euro (e.g. 24.7 will be rounded down to 24).

Write, compile and test whether the application provides the right change. Save the class as `ChangeCalculator.java`

### Problem 6

Write an interactive java program that accepts the radius of a circle from the user as input. The program computes and displays the area of the circle. Use a constant in the program. Save the program as `CircleArea.java`.

*Hint:* the formula to compute the area of a circle is:

$area = \pi * radius * radius$

## 2 Debug Java programs

Using the demo on how to debug a Java program shown during class, debug the programs from the *Debug* folders located on Moodle under *Topic2*. Download the files from the *Debug* folder. Compile each file, try to solve one compilation error at a time, save the file, compile the file again; repeat the same process until the program does not contain any compilation errors. Run the programs.

- Download the file `DebugTwo1.java` and fix any errors it contains
- Download the file `DebugTwo2.java` and fix any errors it contains
- Download the file `DebugTwo3.java` and fix any errors it contains

- Create at least one entry in the *Errors glossary* (available on Moodle under the *Forum news*) corresponding to one of the errors that you corrected while debugging any of the above programs. If the same error has already been recorded in the *Errors glossary* add a comment about your particular solution.