

## Sheet 1: Java Basics – Proposed Solution

### Eclipse

#### Open Eclipse on ZID Computers

**TL;DR** `module load eclipse/4.7.2/java && eclipse`

**Long Version** The ZID provides Eclipse in form of a module<sup>1</sup>. This means that you first have to check which version of Eclipse they currently provide. To do this execute `module avail` in the terminal of the ZID computer. Select your preferred version (e.g. the latest) and load that module. Therefore execute `module load eclipse/4.7.2/java` which loads the Eclipse module. After that enter `eclipse` to start Eclipse from the terminal.

#### Preparation Task

Create a new Eclipse project and import the given Java file `MyFirstProgram.java`<sup>OLAT</sup>. Any file that is formatted like `MyFirstProgram.java`<sup>OLAT</sup> is provided in the OLAT<sup>2</sup> course of the Proseminar. After doing so, start answering the questions given below in the discussion part of this sheet. Do not hesitate to ask questions if something is not clear. Use the opportunity to ask in the discussion group as well as the forum in OLAT to get help. If something remains unclear after solving the exercises feel free to write a plain text file that you include in your submission.

#### Discussion Part (solve in the PS; no submission needed)

- a) ☐ ★ Run the previously added program by clicking `Run As` > `Java Application` in Eclipse. Eclipse will automatically execute the `main` method and the contained code. In this example program the output will be "Hello, World!" followed by some lines created within a loop.
- b) ☐ ★★ Create a new class called `Fibonacci`<sup>3</sup> that includes a `main` method. This method should compute the first 10 Fibonacci numbers. Please use an **iterative** approach to compute these number with a `while` loop.

#### Solution



`Fibonacci.java`<sup>OLAT</sup>

<sup>1</sup>[https://en.wikipedia.org/wiki/Environment\\_Modules\\_\(software\)](https://en.wikipedia.org/wiki/Environment_Modules_(software))

<sup>2</sup><https://lms.uibk.ac.at/auth/RepositoryEntry/4457594910/CourseNode/99190779272774>

<sup>3</sup>[https://en.wikipedia.org/wiki/Fibonacci\\_number](https://en.wikipedia.org/wiki/Fibonacci_number)

- c) ★ You can use breakpoints to debug your code. Rightclick on the left border of the editing window in Eclipse (gray area left of the code next to the line numbers)<sup>4</sup> and select “Toggle breakpoint”. As an alternative you can use the keyboard shortcut `Ctrl + Shift + B`. To execute your program with the debugger click `Run >> Debug as >> Java Application`.

Eclipse will now switch to the debugging view. You will be able to see the code, the contained variables, their current values and the previously set breakpoints. You can run the code “step-by-step” by using the keyboard shortcuts `F5`, `F6` or `F8`. Please take a look at the file `Debug.java`<sup>OLAT</sup> and try to determine the value of `x` in the third iteration of the loop by using the debugger.

#### Solution



`x = 2.5`

- d) ★★ Try to answer the following questions:

- What is the difference between a class and an object?
- What is the difference between a method and a function (i.e. in C)?

#### Solution



- A class is a template that is used to create objects that are instances of that class.
- Methods exist in the context of a class in contrast to functions. This means that methods can access the internals of a class in addition to its arguments, while a function normally gets all data through arguments.

In Java there are only methods as the concept of a function does not exist.

## Homework Part (solve at home; submission required)

### Exercise 1 (Internal Operators)

[3 Points]

Take a look at the following expressions and operations and give their result as well as their data type and an explanation how these results are computed. You are advised to write a `main` method to test these expressions. Either use `System.out.println()` or the debugger to get the results. If some expressions are invalid give a reason why this is the case.

**Submit:** a plain text file containing all your answers.

<sup>4</sup>By default the line numbers are disabled in Eclipse. You can enable them by rightclicking on the gray area and then selecting “Show Line Numbers” in the opened menu.

## Hint



Only submit **UTF-8** encoded text files. Use a modern text editor or Eclipse to create that file. Some recommended editors are VIM<sup>a</sup>, Emacs<sup>b</sup>, Sublime<sup>c</sup>, Atom<sup>d</sup>, gedit<sup>e</sup> or Notepad++<sup>f</sup>.

<sup>a</sup><http://www.vim.org>

<sup>b</sup><https://www.gnu.org/software/emacs/>

<sup>c</sup><https://www.sublimetext.com/>

<sup>d</sup><https://atom.io/>

<sup>e</sup><https://wiki.gnome.org/Apps/Gedit>

<sup>f</sup><https://notepad-plus-plus.org/>

Expression	Result
6 * 5 / 3	10 (int)
1 << 8 % 3	4 (int)
(short) Integer.MAX_VALUE	-1 (short) → Cast führt zu Overflow
23 / (double) 11	2.090909090909091 (double)
(double) (23/11)	2.0 (double) → Integer-Division rundet ab
42f	42.0 (float)
4e3D	4000.0 (double) → Scientific Notation
11 * 1.2 != 47	true (boolean)
"Peter=Coffee+" + 'chocolate' + 2.0	SyntaxError → Single Quotes nur für chars
"Peter=Coffee+" + "chocolate" + 2.0	Peter=Coffee+chocolate2.0 (String)
1 == 24 % 3 && 4 > 7    true	true (boolean)
1 == 24 % 3 ? 4 : 7	7 (int)

## Exercise 2 (for, if, switch, Arrays)

[3 Points]

The programm contained in `ShoppingCart.javaOLAT` simulates a shopping cart. It contains prices for items that are added. You have to implement some discount functionality. First, there is a discount on the total price. Second, there is a discount based on the number of items in the cart.

## Hint



It is no problem, if you do not fully understand the given code as some of the used concepts did not come up in the lecture yet. I.e. the meaning of `static` and `private` will be explained in later lectures.

These two tasks are described in detail in the following:

- 2 Points** Read the given code of the class carefully. Give special attention to the TODO's and update the code as requested. When done, test your code by running the `main` method. Do not change any code that is not marked with a `// TODO` comment. You therefore have to add some output functionality and the discount calculation only. Do not use any advanced concepts of Java (i.e. `ArrayList`) for your solution. Please document your code using comments.
- 1 Point** When you take a look at the top of the `main` method you will see four different carts. Test your implementation with all four predefined shopping carts (`shoppingCart1`, `shoppingCart2`, `shoppingCart3` and `shoppingCart4`). Does your program work with all of them? Where are the differences in execution? E.g., which branches of the code are executed?

**Submit:** your updated Java class file and a plain text file containing the console outputs for all four carts.

## Solution



ShoppingCartSolution.java<sup>OLAT</sup>

## Exercise 3 (Objects and Classes)

[4 Points]

- a) **1 Point** Create a new class called `Person`. This class contains a `firstname (String)` and a `lastname (String)`. Further, you need to add getters and setters for these two attributes<sup>5</sup>.
- b) **0.5 Points** Implement a constructor that initializes both previously created attributes<sup>6</sup>. In addition, create a method that prints the full name of a person on the console.
- c) **0.5 Points** Create a `main` method and do the following:
- Create two persons and name them.
  - Print both persons on the console.
- d) **2 Points** Extend your class `Person` with a method `greet` that takes an object of the `Person` class as parameter. This method prints a greeting text towards the person given as a parameter.

Include the following code snippet in the `main` method to test your implementation.

```
1 Person john = new Person("John", "Doe");
2 Person donald = new Person("Donald", "Duck");
3 john.greet(donald);
```

The printed message on the console has to be: John Doe says "Hi!" to Donald Duck.

**Submit:** your Java class file and a plain text file containing the console output.

## Solution



Person.java<sup>OLAT</sup>

**Important:** Submit your solution (.txt, .java or .pdf) to OLAT and mark your solved exercises with the provided checkboxes. The deadline ends at 23:59 on the day before the discussion.

<sup>5</sup>To generate them in Eclipse either click `Source >> Generate Getter and Setters` or use the keyboard shortcut `Alt + Shift + S` followed by `R`.

<sup>6</sup>To generate a constructor in Eclipse click `Source >> Generate Constructor using Fields` or use the keyboard shortcut `Alt + Shift + S` followed by `C`.