



Informatics I – EProg HS15

Exercise 1

1 Task: Hello World

1.1 Learning Objectives

1. Install Java Development Kit (JDK)
2. Create, compile, and execute Java classes

1.2 Assignment

a) Install JDK and Editor

Download the current version of the [Java SE \(Standard Edition\) Development Kit \(JDK\)](#) and install it¹. If you have troubles during the installation [this tutorial](#) may help. You can find additional information about the installation on Windows systems [here](#). Afterwards, install any text editor with syntax highlighting².

b) Create and execute Program

Copy the code for the two classes listed below into two separate text files and store them under the name <classname>.java (i.e. HelloWorld.java). Use one of the suggested text editors or an editor of your choice. Compile the two classes and execute the test driver³ HelloWorldTest.

```
1 public class HelloWorld {
2
3     /**
4      * Simply prints "Hello World" to the console.
5      */
6     public void printHelloWorld() {
7         System.out.println("Hello World!");
8     }
9 }
```

Listing 1: HelloWorld class

¹The current version is JDK 8 Update 60 (8u60). Mac OS X already includes a pre-installed JDK

²E.g. [JEdit](#), [Sublime Text](#), [Notepad++](#), [Vim](#), [Emacs](#), [TextWrangler](#), etc.

³A test driver is a class that implements a `main()`-method (entry point to each program) and whose main purpose is to test objects of classes

```

1 public class HelloWorldTest {
2
3     /**
4      * Entry point for a Java program. Creates a HelloWorld object and
5      * calls printHelloWorld() on it.
6      */
7     public static void main(String[] args) {
8         HelloWorld myHelloWorldObject = new HelloWorld();
9         myHelloWorldObject.printHelloWorld();
10    }
11 }

```

Listing 2: Test driver HelloWorldTest for testing the HelloWorld class

1.3 Remarks

1. If you have problems solving this exercise, consider doing the online module of Task 4 first.

Solution:

```

compile program:
javac HelloWorld.java HelloWorldTest.java
execute program:
java HelloWorldTest

```

a) Programm Modification

Let the program print another text than "Hello World!".

2 Task: Cuboid

2.1 Learning Objectives

1. Read and comprehend source code
2. Test programs with a test driver
3. Be able to change program logic

2.2 Assignment

a) Create Cuboid

Copy the code below into a separate text file as you did in task 1. The variables `length`, `width`, and `height` store the respective attributes of a cube. The method `printVolume()` computes the volume of this cube and prints it to the console. Your task is to create a test driver to test this behavior.

```
1  /**
2   * Represents a cuboid.
3   */
4  public class Cuboid {
5      private int length = 50;
6      private int width = 30;
7      private int height = 10;
8
9      /**
10     * Calculates the volume of this cuboid
11     * and prints the result to the console.
12     */
13     public void printVolume() {
14         System.out.print("The volume of this cuboid is ");
15         System.out.println(length * width * height);
16     }
17 }
```

Listing 3: Cuboid class

Solution:

```
1  public class CuboidVolumeTest {
2
3      /**
4       * Entry point for a Java program. Test a Cuboid object in it.
5       */
6      public static void main(String[] args) {
7          Cuboid cuboidObject = new Cuboid();
8          cuboidObject.printVolume();
9      }
10
11 }
```

Listing 4: Cuboid test driver

b) Program Modification

After you have successfully executed the program, try to extend it such that it is able to compute and print the surface of the cube (analogous to the volume). Hint: both the `Cuboid` class and test driver need to be adapted.

Solution:

```
1  /**
2   * Represents a cuboid.
3   */
4  public class Cuboid {
5      private int length = 50;
6      private int width = 30;
7      private int height = 10;
8
9      /**
10     * Calculates the volume of this cuboid
11     * and prints the result to the console.
12     */
13     public void printVolume() {
14         System.out.print("The volume of this cuboid is ");
15         System.out.println(length * width * height);
16     }
17
18     /**
19     * Calculates the surface of this cuboid
20     * and prints the result to the console.
21     */
22     public void printSurface() {
23         System.out.print("The surface of this cuboid is ");
24         System.out.println(2*length*width + 2*length*height + 2*width*
25                             height);
26     }
27 }
```

Listing 5: Modified Cuboid class

```
1  public class CuboidSurfaceTest {
2
3      /**
4       * Entry point for a Java program. Test a Cuboid object in it.
5       */
6      public static void main(String[] args) {
7          Cuboid cuboidObject = new Cuboid();
8          cuboidObject.printSurface();
9      }
10 }
```

Listing 6: Modified test driver

3 Task: Code Comprehension

3.1 Learning Objectives

1. Comprehension of primitive data types
2. Understand arithmetic expressions and type conversion
3. Practice reading and comprehension of source code

3.2 Assignment

a) Type Conversion

Values of which data types can be assigned to a variable of another data type without the need of explicit type casting? Complete the following assignment-compatibility table:

byte ->	->	->	->	-> double
char ->	->	->	-> double	

Solution:

byte -> short -> int -> long -> float -> double
char -> int -> long -> float -> double

b) Arithmetic Expressions and Operations

1. Complete Table 1:

Mathematic expression	Expression in Java
$rate^2 + delta$	
$\frac{10-a}{bc}$	
=	

Table 1: Arithmetic expressions in Java

Solution:

Mathematic expression	Expression in Java
$rate^2 + delta$	rate * rate + delta
$\frac{10-a}{bc}$	(10 - a) / (b * c)
=	==

Table 2: Solutions to Arithmetic expressions in Java

2. Show three different ways to subtract 1 from an integer (`int`) variable.

Solution:

- (a) `i--` (or `i++`)
- (b) `i = i - 1`
- (c) `i -= 1`

c) Code Snippets

Which of the following eight code snippets are syntactically correct and what is the output in those cases? Motivate your answers and especially explain unexpected results.

```
1  int x = 20;
2  int y = 10;
3  System.out.println(x + y++);
4  System.out.println(y);
```

Listing 7: Snippet 1

Solution:

correct
Output: 30 and 11
Motivation: `y` is incremented after the evaluation of the expression.

```
1  byte x = 125;
2  byte y = 5;
3  x = (byte)(x + y);
4  System.out.println(x);
```

Listing 8: Snippet 2

Solution:

correct
Output: -126
Motivation: the range of `byte` is only -128 to +127.
1000 0010 in the case of `int` represents 130
1000 0010 as two's complement in the case of `byte` therefore corresponds to -126

```
1  short x = 128;
2  byte y = (byte)x;
3  System.out.println(y);
```

Listing 9: Snippet 3

Solution:

correct
Output: -128
Motivation: the range of byte is only -128 to +127.

```
1 byte x;  
2 short y = 120;  
3 x = y;  
4 System.out.println(x);
```

Listing 10: Snippet 4

Solution:

not correct
Output: does not compile.
Motivation: no implicit type conversion from short to byte possible.

```
1 short x;  
2 byte y = 120;  
3 x = y;  
4 System.out.println(x);
```

Listing 11: Snippet 5

Solution:

correct
Output: 120
Motivation: implicit type conversion is possible.

```
1 char t = 't';  
2 System.out.println((long)t);  
3  
4 t = 84;  
5 System.out.println(t);  
6  
7 t = '9';  
8 System.out.println((int)t);
```

Listing 12: Snippet 6

Solution:

correct
Output: 116, T, 57
Motivation: output of the ISO char values (116, 57) and the Characters (T), respectively.

```
1 System.out.println(1/4);  
2 System.out.println(1/4d);  
3 System.out.println((double)(1/4));  
4 System.out.println((double)1/4);  
5 System.out.println(1.0/4);
```

Listing 13: Snippet 7

Solution:

correct.
Output: 0, 0.25, 0.0, 0.25, 0.25
Motivation: in the first line: `int` values are used per default. In the third line: the resulting `int` is converted into a `double`.

```
1 double x = 3;  
2 double y = 29;  
3 System.out.println(x / (y % ++x));
```

Listing 14: Snippet 8

Solution:

correct.
Output: 3.0
Motivation: `x` is incremented before the expression is evaluated.

4 Task: Online Module

4.1 Learning Objectives

1. Learn fundamental terms of software development and understand the basic development process
2. Being able to write, compile, and execute a simple Java program
3. Understand and correct simple compilation errors

4.2 Assignment

1. Work through the eLearning Module 1 in OLAT. To do so, register for the course *INF_15_HS Informatik I* and click on the eLearning link in the navigation bar.
2. After finishing the self-study, solve the self test to check your knowledge.