PROGRAMMING 1 / Written Assignments

The objective of these weekly written assignments is to guide you to ***systematically familiarise yourself with the course materials*** and help you to ***learn fundamental concepts more deeply***.

Please write your weekly ***short*** answers to separate Word documents called

SWD32_Week_*n_yourSurname*.docx where *n* is the week number.

* Submit the Word documents to **Moodle.** Alternatively, you can submit PDF files.

Week 1

1.1 What is the ***purpose of the main method***?

1.2 Java is a *strongly typed programming language*. What does ***strongly typed*** mean here?

1.3 What if there is a ***syntax error*** in your Java code, ***does your program crash due to the syntax error***? Give arguments!

1.4 There are 12 symbols in the Java code below.

```
if (x > 0) {
    y = 1;
}
```

What ***identifiers, keywords, literals, operators***, and ***separators*** you can find in the code?

1.5 Write the following single-line comment: **`// My comment`**

a) as a ***multi-line comment***

b) as a ***Javadoc comment***

1.6 What does ***promotion*** mean in Java?

1.7 Explain why you can or cannot successfully assign

a) the value of **1_999_999_999** to a variable of type ***int***?

b) the value of **2_999_999_999** to a variable of type ***int***?

c) the value of **2_999_999_999** to a variable of type ***double***?

d) **"1"** to a variable of type ***int***?

e) **'A'** to a variable of type ***String***?

f) **"A"** to a variable of type ***char***?

g) **"A"** to a variable of type ***String***?

1.8 What is wrong in the code below? **Give arguments**!

```
final int x = 100;
x++;
```

1.9 What is ***wrong*** in the code below? **Give arguments**!

```
for (int i = 1; i <= 5; i++) {
    System.out.println(i);
}

System.out.println(i);
```

1.10 What does the code below print? You can run the code in Eclipse to see the results.

NB! The objective is to understand ***why*** Java gives certain results. Therefore, **give arguments**!

```
System.out.println( 4 * 2 / 4 );          // a)
System.out.println( 2 / 4 * 4 );          // b)
System.out.println( 2.0 / 4 * 4 );        // c)
System.out.println( (int)0.99 );          // d)
System.out.println( 3 * 1.1 );            // e)
System.out.println( 1 < 2 );              // f)
```

Week 2

Control structures

2.1 What is ***wrong*** in the code below?

```
int i = 0;
while (1 < 10) {
    System.out.println(i);
}
```

2.2 What is the *standard way to write a clean while loop*?

2.3 By default, the program crashes when the code line below is executed.

```
int age = Integer.parseInt("ten years");
```

Explain how we can ***prevent the program to crash*** when the code line above is executed. Assume that we do not modify the code line.

Methods

2.4 What does ***passing parameters by value*** mean?

2.5 How does ***passing arrays as parameters*** differ from ***passing ints*** as parameters?

2.6 What is the purpose of the keyword ***void***?

2.7 What does ***method overloading*** mean?

2.8 What are the ***two given rules of thumb for limiting the size of a method***?

2.9 What does '***method library class'*** mean? How does a method library class differ from a '***program class'***? What kind of benefit we can get by creating and using 'method library classes'?

2.10 In your program, what are the ***two possible ways to use a class from another namespace (package)***?

Week 3

Arrays

3.1 How do arrays in Java and 'arrays' in JavaScript differ from each other?

3.2 What is ***wrong*** in the code below?

```
int[] array = {4, 3, 1, 6};

System.out.println(array[4]);
```

3.3 What does the method `Arrays.copyOf` do?

Internal representation of primitive types

3.4 What are the internal representations (binary representations) of the values in the variables below?

```
byte a = 3;
int x = 15;
int y = -1;
char c = 'c';
```

3.5 What does the code below print? ***Explain why the specific values are printed.***

```
System.out.println( 1 + 16 );

System.out.println( "1" + 16 );

System.out.println( '1' + 16 );

System.out.println( (char)('1' + 16) );
```

Week 4

String handling

4.1 What is ***wrong*** in the code below?

```
System.out.println('Hello!');
```

4.2 What does '*strings are immutable*' mean?

4.3 What is ***wrong*** in the code below?

```
Scanner input = new Scanner(System.in);

System.out.print("Enter your name: ");
String name = input.nextLine();
if (name == "") {
    System.out.println("You did not enter anything");
}
```

4.4 What is ***wrong*** in the code below?

```
String city = "Helsinki";

city.toUpperCase();
System.out.println(city);
```

4.5 What does the code below print?

```
String text = "1,2,3,4";
String[] itemArray = text.split(",");

for (String item : itemArray) {
    System.out.println(item);
}
```

Regular expressions

4.6 Explain how the two operations below differ from each other.

```
text.equals("[1-9]")

text.matches("[1-9]")
```

4.7 The method call below returns **true**, when the text is either "Java" or "JavaScript".

```
text.matches("Java(Script)?")
```

Explain why the method call below returns ***true*** when the text is "Java" and ***false*** when the text is "JavaScript".

```
text.matches("Java[Script]?")
```

4.8 Explain why ***two backslashes*** are written below.

```
text.matches("(What|Why|When)\\?")
```

Week 5

5.1 What kind of ***class members*** a typical class for objects has?

5.2 What does the ***new*** operator do?

5.3 What is the purpose of the ***toString*** method?

5.4 What is the purpose of a ***getter*** method?

5.5 What is the purpose of a ***setter*** method?

5.6 What is the purpose of a ***constructor***?

5.7 What is meant by ***reference variable***?

5.8 What is the main difference between ***primitive types*** and ***reference types***?

5.9 What does ***reference copy*** mean in Java?

5.10 How can we promote ***encapsulation*** in Java?

Week 6

6.1 What is meant by ***constant*** in Java?

6.2 What is meant by ***garbage collection***?

6.3 What is meant by ***inheritance***? Explain very shortly the basic idea only.

6.4 Why do we need ***primitive wrapper classes***?

6.5 What is meant by ***autoboxing***?

6.6 How do ***class fields*** and ***instance fields*** differ from each other?

6.7 How do ***class methods*** and ***instance methods*** differ from each other?

6.8 How many objects Java creates when the code below is executed? **Give arguments**!

```
Stopwatch[] timerArray = new Stopwatch[3];
Stopwatch timer = new Stopwatch();
Stopwatch sw;
sw = timer;
```

Week 7

7.1 Suppose the following code:

```
public interface Singer {
    public void sing();
}

public class Guitarist implements Singer {
    public void sing() {
        System.out.println("la-la-la");
    }

    public void play() {
        System.out.println("twang");
    }
}
```

What is wrong in the piece of code below? There is one error.

```
Singer mo = new Guitarist();
Guitarist ed = new Guitarist();

mo.sing();
mo.play();
ed.sing();
ed.play();
```

7.2 What does the code below print?

```
int[] myArray = { 3, -1, 0, 4, 3 };

Arrays.stream(myArray)
    .filter(x -> x > 0)
    .distinct()
    .forEach(x -> System.out.println(x));
```

7.3 Suppose that we sort the array below with the ***insertion sort*** algorithm.

```
int[] myArray = { 4, 5, 2, 3 };
```

The initial order of the values is the following: 4, 5, 2, 3.

What is the order of these values in the array ***after each iteration of the outer loop*** of the insertion sort algorithm?

7.X OPTIONAL task for advanced learners

Modify the Java code of the insertion sort algorithm so that it sorts an array of ints in ***descending order***. NB! The objective is to understand the original code and practice your algorithm design skills. Therefore, just change the sorting order within the original algorithm. Do not add extra steps to the algorithm or use any external methods.