Programming Basics – live exercises

Introduction

Task 1: Understanding terminology

Decide whether the following statements are true or false.

	True	False
In contrast to problem-oriented programming languages, machine-oriented programming languages are always tailored to a computer platform.	✓	
You can use the same compiler on all platforms.		✓
If the source code is written in a machine-oriented programming language, then the translator is called a compiler.		✓
Processors can directly run programmes written in problem-oriented programming languages.		✓
A programme is a form of operating instructions for a computer system.	✓	
Compiled programmes make optimal use of the properties of the respective platform.	✓	
Compilers convert a programme from a source language into an equivalent programme in a target language.	✓	
Programmes translated by a compiler are platform dependent.	✓	
The task of a compiler is to convert programmes in machine language into programmes that can be read by a programmer.		✓
An algorithm is more accurate than the natural language and more detailed than the programme to be created.		✓
Machine-oriented programming languages can be used anywhere, regardless of the platform.		✓

Task 2: Fill in the gaps

Fill in the gaps in the following text by adding one or two words in the marked areas.

(1) In a programming language, both the notation (synonym: syntax) as well as the meaning (synonym: semantics) of the individual instructions must be defined.
(2) The task of a compiler is to convert all sentences in a source language into equivalent sentences in a target language.
(3) In the case of programming languages, a distinction is made as to whether they can be compiled or interpreted or both.
(4) When programming, the problem solution is often first written down in the form of an algorithm. An algorithm is semi-formal, i.e. in contrast to the natural language it's described in more detail, but not yet fully detailed, as is required by a programming language.
(5) In order for Java programmes to be platform independent, they run within a virtual machine, which is also called a Java virtual machine. For all common operating systems, this is provided in the form of the Java Runtime Environment.
(6) An applet is a Java bytecode programme that runs in a web browser.
(7) A servlet is a Java bytecode programme that runs on a web server.
(8) An application is an executable Java bytecode programme that does not require a web browser.
(9) Complete the programme so that it outputs "Good luck!" on the screen.

```
public class Output {
  public static void main(String[] args) {
    System.out.println(" \"Good luck!\" "); // with quotation marks
    System.out.println("Good luck!"); // without quotation marks
  }
}
```

(10) The name under which the programme's source file is stored by (9) should be Output.java.
(Please make sure to use the correct upper/lower case).
(11) To compile the Output.java file in the DOS window, the following command must be entered: javac Output.java
(12) The compiler creates a file that contains bytecode. The file created in (11) will have the name Output.class.
(13) To execute the bytecode from (12), enter the command java Output in the DOS window.
(14) The programme that interprets the bytecode is called Java virtual machine (JVM) in Java.
(15) A text editor can be used to create a Java source file.
(16) The main method is the point at which a programme is started.
(17) A comment is a note about the programme intended for humans.
(18) Java distinguishes between three types of comments: (i) line comment,
(ii) block comment and (iii) Javadoc comment.

Task 3: Syntacticly Correct?

Decide whether the following code snippets have correct syntax or not. If you find any errors, please mark them in the source code.

(1) Syntax is incorrect – a closing bracket is missing.

```
public class MyClass
{
  public static void main (String[] args)
  {
    System.out.println("Hello!");
}
```

(2) Syntax is correct - however, it contradicts the rules for easy-to-read code, because there should only be one instruction (statement) per line

```
public class MyClass
{
  public static void main (String[] args)
  {
    System.out.println("Hello"); System.out.println(" World! ");
  }
}
```

(3) Syntax is incorrect – the keyword class is missing

```
public MyClass
{
  public static void main (String[] args)
  {
    System.out.println("Hello!");
  }
}
```

(4) Syntax is incorrect – the main method must not end with a semicolon

```
public class MyClass
{
  public static void main (String[] args);
  {
    System.out.println("Hello!");
  }
}
```