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

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 | prog.... 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…idea…… . An …Algorithm…….. is semi-formal, i.e. in contrast to the …colloquial language… it's described in more detail, but not yet fully detailed, as is required by a …compiler…………………… .

(5) In order for Java programmes to be platform independent, they run within a ……JVM……………, 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!\"");
  }
}
```

(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 …………………………….. 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 …interpreter…….. 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)

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

(2)

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

(3)

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

(4)

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