Assignment 2 - Theoretical Questions

1. What is the Java Virtual Machine? What is Bytecode?
    - Answer: JVM is a Virtual Machine that enables a computer to run Java programs. The Java bytecode is what you can call the instructions set of the JVM.

2. What is the Java Classpath?
    - Answer: The Java classpath is a parameter in the JVM that specifies the location of the users classes and packages

3. How do you compile and run your java program without the help of an Integrated Development Environment (IDE) (e.g., an IDE like Eclipse)?
    - Answer: You can compile a java program using the command line, assuming you have JDK installed. In Linux for example, you can write “javac -source 1.4 -target 1.4 helloworld/HelloWorld.java”, where 1.4 is the version of the JDK.

4. What is a JAR file?
    - Answer: JAR stands for Java Archive and is a package file format. These files are used to efficiently deploy a set of classes and their associated resources.

5. How do you declare the starting point of a Java application?
    - Answer: The starting point of a Java application is the main() method of a class. It is in this method the program starts. (public static void main(String[] args{ … }).

6. What is a package? Why is important to declare classes inside packages?
    - Answer: A package is a namespace that organizes a set of related classes and interfaces. Having classes in packages makes them go under the same namespaces and this also prevent class name collisions.

7. What is an *interface*? Why is it important to not change them?
    - Answer: An interface is a reference type. It is similar to a class but it is a collection of abstract methods. A class implemets an interface and thereby inherit the abstract methods of the interface. This is also why we should not change an interface.

8. Which visibility levels are available in Java? What is the default visibility for classes, methods, and fields?
    - Answer: There are four levels of visibility. Public, Protected, Default/No modifier and Private. The default one is No modifier which is package private, meaning the class, method etc, can only be accessed within it’s own package.

9. In the context of Java, what is an Exception? And what is an Error?

Kim Nilsson
kn222gp
Utvecklare av Digitala Tjänster
Linnéuniversitetet

- Answer: An exception is a problem that arises during the execution of a program. When an Exception occurs the normal flow of the program is disrupted and the program terminates. Exceptions should be caught and handled. An error on the other hand is a subclass of Throwable that indicates serious problems that a reasonable application should not try to catch. Most of such errors are abnormal conditions.

10. What happened if your program terminates with an *OutOfMemoryError*, or *NoClassDefFoundError* or *NullPointerException*?
    - Answer:
      OutOfMemoryError - No more memory could be made available by the garbage collector.
      NoClassDefFoundError - Happens when the JVM tries to load the definition of a class and no definition of the class can be found.
      NullPointerException - This exception is thrown when a program tries to use null in a case where an object is required.

11. How do you handle Exceptions in your program?
    - Answer: Exceptions should be handled with try/catch.

```
try {
   //Protected code
}catch(ExceptionType e) {
   //Catch block
}
```

12. Why is it important to test your code/application/product, before you deliver it to your customer/boss/teacher?
    - Answer: It's important to test your application before delivery to prevent bugs and check that the application actually fulfill the requirements. Simply, to deliver a product with high quality.

13. What is JavaDoc? How do you write documentation with it?
    - Answer: JavaDoc is a program that comes with JDK (Java Developer Kit) that can be used to produce documentation of your classes. To use javadoc on your source code, you have to tag your code with a certain type of comment formats. Example of this:

```
public class MyButton
{
/**
*
* @param image the image to show on button
* @param label text to show on button
*/
public MyButton(String label, Image image)
{
//code here
}
```

Kim Nilsson
kn222gp
Utvecklare av Digitala Tjänster
Linnéuniversitetet

```
//rest of class here
}
```

Javadoc comments start with /**, end with */, and, in-between, use tags such as @param, @return, and @exception to describe the workings of a method.

Code example taken from http://www.devx.com/tips/Tip/13579

Kim Nilsson
kn222gp
Utvecklare av Digitala Tjänster
Linnéuniversitetet