1. What is the Java Virtual Machine? What is Bytecode?

It basically is the program in which execute programs written in Java on a computer. It is as the name applies a virtual machine and not a physical computer. This also means that the virtual machine suffers from not being able to use the full power of the cpu. Bytecode in java is instructions set of the JVM.

2. What is the Java Classpath?

Classpath is the parameter in JVM or in the java compiler in which specifies the location or path of the user-defined classes or packages in the program. This parameter may be set via command-line or an environment variable.

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)?

You can use a command line/prompt to compile and run programs in java. First it has to be converted to a form that the JVM can understand, this so that all computers using JVM can interpret and run the selected program.

4. What is a JAR file?

It is a package file format used to aggregate several java class files and associated metadata and other resources into one file. This to be able to distribute software or libraries on the java platform. It can be compared to as a archive file like the .Zip.

5. How do you declare the starting point of a Java application?

You declare the starting point of a java application by providing a public static void main method, which takes a String array as an argument.

6. What is a package? Why is important to declare classes inside packages?

A package is a grouping of related types providing access protection and name space managment. Packages in java are used for avoiding naming conflicts, to control access and to make types easier to find.

7. What is an *interface*? Why is it important to not change them?

An interface is a group of related methods with empty bodies. It's the behaviour of something, ex. The behaviour of a bicycle. It's like a skeleton to work/code against. If you have classes that inherits from an interface all these classes needs to be modified or they will break, which it's important not to change them.

8. Which visibility levels are available in Java? What is the default visibility for classes, methods, and fields?

Visibility levels are, visible to the package(this is the default) which is no modifier, visible to the class(private), visible to the “world”(public), vissible to the package and not all subclasses and not visible to the “world”(protected).

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

An error is something severe in java which will make the program crash rather than handle the problem it should not be catched. An exception is a throwable and will tell you what the problem is and will suggest what has to be done to fix it.

10. What happened if your program terminates with an *OutOfMemoryError*, or *NoClassDefFoundError*or *NullPointerException*?

OutOfMemoryError will make the program to crash, this is thrown because the JVM couldn’t allocate an object because it is out of memory to continue. This also means that the garbage collector could not create more memory.

NoClassDefFoundError is an error which is thrown if the JVM or a ClassLoader instance tries to load in the definition of a class and if there is no definition of this class.

NullPointerException is thrown when an application attempts to use null in a case where an object is required.

11. How do you handle Exceptions in your program?

You should and could use Try Catch statements to make the program throw exceptions it there is a problem.

12. Why is it important to test your code/application/product, before you deliver it to your customer/boss/teacher?

Because there is probably some functionality expected by the customer and you need to test these so that they work correctly and so that errors and exceptions are handled correctly. It’s also required to keep the application at a certain standard by testing it.

13. What is JavaDoc? How do you write documentation with it?

Javadoc is an automatic documentation generator for the modern java language which generates API documentation in HTML format based on the source code.

You use a comment structure of multi-line comments that the generator will look for. Then the first paragraph should be a description of the used method. This is followed by a varying number of description tags like, parameters being used, the return value of the method and exceptions that may be thrown in the method. You can also use the less-common “see” tag if there is something else that needs to be noted.