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

The java virtual machine is, as the name suggests, a virtual machine that enables a computer to run a java program. JVM reads the bytecode that is created when you program and compile a java program. Java bytecode is the instruction set of the JVM.

2. What is the Java Classpath?

The classpath is a parameter in the JVM that specifies the location of user-defined packages and classes. It can be set on the command line or through the 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 compile and run a java program through the command line as long as you have a jdk installed. The java compiler is used by typing “javac YourProgram.java”. Then you can run the program by typing “java YourProgram”.

4. What is a JAR file?

A JAR file is a package file format used to aggregate many Java class files, metadata and resources into one file. The java archive file is used to package and distribute your application software or libraries on the Java platform.

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

By using a public and static main() method that takes a string array as parameter.

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

A package is a technique to group related types and providing access protection and namespace management. It is important to use in order to avoid naming conflicts, to control access and to make classes easier to find and use.

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

An interface is a group of related methods with empty bodies. You should not change interfaces as they could be used by multiple classes. If changed other classes may break.

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

Public, protected, private and no modifier(or package private). The default is no modifier.

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

Exception is a form of Throwable that indicates what conditions a reasonable application might want to catch.

An error is a form of Throwable that might be thrown during execution but not caught. Most applications should not try to catch an error. Most errors should be thrown due to abnormal conditions.

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

If your program terminates with an OutOfMemoryError the error was thrown because the JVM could not allocate an object because it is out of memory and the garbage collector could not make more available.

The NoClassDefFoundError is thrown when the JVM or a ClassLoader instance tries to load in the definition of a class and no definition of that class could be found.

The NullPointerException is thrown when an application tries to use null in a case where an object is required. For example when calling the instance method of a null object.

11. How do you handle Exceptions in your program?

Using try-catch statements to catch the exceptions that are thrown in the program.

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

To make sure the code/application/product works as it should.

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

JavaDoc is a documentation generator created by Sun Microsystems for Java used for generating API documentation in HTML format from java source code. You write JavaDoc documentation by using multi line comments. The first paragraph should be a description of the method. Following the desciprion there should be a number of descriptive tags describing the parameters with @param, what the method returns with @return) any exceptions that may be thrown with @throws and lastly the see also @see for other information.