ASSIGNMENT 2

Q.1. What is the difference b/w JDK, JRE & JVM ?

JDK	JRE	JVM
1. To compile the java code, we need a tool and that tool is tool is called JDK (Java Development Kit)	JRE stands for Java Runtime Environment. JRE contains all the built in classes and other files and libraries that we can use in our code	JVM stands for Java Virtual Machine Java is a space where you execute the code
2. Byte code runs on a machine so we have to convert the java code into bytecode	JRE validates the bytecode and loads a class	JVM makes java language platform independent, You have to compile through the JDK and run through JVM
3. On installing the JDK we get both updated JRE and JVM	If we use extra Libraries & also you need an environment. An additional layer is present above the OS known as JRE	JVM is the platform dependent and it needs to be build for particular OS

Q.2 What is JIT Compiler?

JIT is the part of JVM that optimizes the performance of the application. JIT stands for JUST IN TIME Compiler.

JIT is the integral part of JVM. It accelerates the execution performance many times over the previous level. In other words, it is a long running, computer intensive program that provides the boost performance environment.

Q.3 What is class loader?

The java classloader is the part of JRE that dynamically loads Java classes into the JVM. The Java Runtime system does not need to known about files & file systems because of class loaders.

Java classloader is of three types:
1. Bootstrap classloader
2. Extension classloader
3. System classloader

Q.4 Explain various memory logical partitions

Memory management in OS is the function responsible for allocating and managing a computer main memory. Memory management function keep track of the status of each memory location, either allocated or free, to ensure the effective & efficient use of Primary Memory.

There are two Memory Management techniques:

1. Contiguous
2. Non Contiguous

In Contiguous Technique, executing process must be loaded entirely in the main memory. Contiguous technique can be divided into:

1. Fixed / static Partitioning
2. Variable / dynamic Partitioning

Fixed Partitioning

This is simple and the oldest technique used to put more than one process in the main memory. In this partitioning, the number of partitioning (non-overlapping) in RAM is fixed but the size of each partition may or may not be the same. As it is a contiguous allocation, hence no spanning is allowed. Here partitions are made before execution or during system configuration

Q.5 What gives Java its "Write Once & Run Anywhere" nature?

Java language is "Write Once & Run Anywhere" nature gets from its bytecode. Java programs are first converted into class files (also known as bytecode), an intermediate language before being converted into machine code. The JAVA code can be written on any device or machine or platform and class file will remain the same throughout. This Java intermediate code can be run on any platform be it Windows, Linux provided the system has JRE & JVM installed to run the bytecode.

Q.6 Explain the history of Java. Who invented class?

Java was conceived by James Gosling, Patrick Naughton, Chris Warth, Ed Frank and Mike Sheridan at Sun Microsystems Inc. in 1991. It took 18 months to develop the first working version. This language was initially called "Oak" but was renamed "Java" in 1995.

Bjarne Stroustrup is a Danish Computer Scientist, most notable for the invention and development of C++ programming language. C++ was an enhancement to the C language, called as C language with classes.

Q.7 What was the original name of Java? Why it was renamed?

The Java language was initially called "Oak" but was renamed in 1995. Oak was already a trademark by Oak Technologies, hence the reason for renaming the language as JAVA.

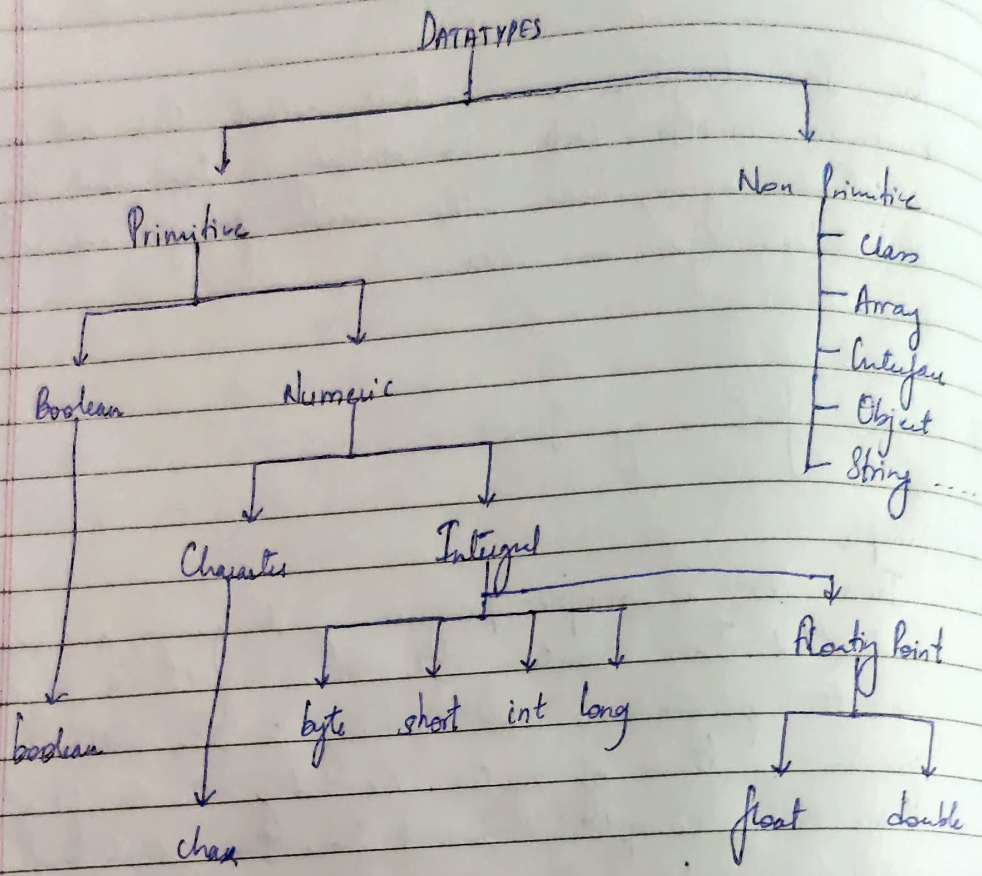
Q.8 List features of Java

1. Simple
Java is a simple programming language and easy to understand. Java contains the same syntax as in C, C++, so the programmers who switching to Java will not face any problem in terms of syntax. Secondly the concept of pointers have been completely removed from Java which leads to confusion for a programmer and pointers are also a vulnerable to security

2. Object Oriented
Java is an object oriented programming language, which means in java everything is written in terms of classes and objects

3. Platform Independent
4. Portable
5. Robust
6. Secure
7. Interpreted
8. Multithreaded

Q.9 List various Datatypes in Java



Q.10 What is the difference b/w System.out.print
System.out.println
System.err.print

System.out refers to the Standard output stream

System.err refers to the Standard error stream, which is also console by default.

System.out and System.err are objects of the type PrintStream

System.out.print() → It is used to print an argument that is passed to it

1. System → It is a final class defined in java.lang package

2. out → This is an instance of PrintStream type, which is a public and static member field of the System class.

3. print() → It prints an argument passed to it.

4. println() → It prints an argument passed to it and adds a new line to the output.

5. err → This is also an instance or object of PrintStream type.

Q.11 How is Java platform independent?

Java is platform independent because of its "write once & run anywhere" nature which comes from its bytecode or class file. This bytecode or class file or intermediate code can be run on any platform.

Q.12 What is bytecode? How is it different from machine code?

Bytecode is highly optimized set of instructions designed to be executed by Java runtime system which is also called Java Virtual Machine (JVM). In essence, the original JVM was designed as an interpreter for the bytecode.

This may come as a bit of surprise since many modern languages are designed to be compiled into an executable code. However the fact that a Java program is executed

by the JVM helps solve the major problems associated with web based programs

Q.13 What is difference b/w jar file and runnable jar file?

A Jar file (Java Archive) is a package file format typically used to aggregate many Java class files and associated metadata and resources (text, images, etc) into one file to distributed application software or libraries on the java platform.

With the standard Jar file, you have to specify the class with the main method on the command line when running the jar

With the runnable jar, there is a manifest file that will hold that information so you can just type java -jar myRunnable.jar or simply double click it

Q.14 What is difference b/w Runnable jar file & exe file?

An exe file is an executable file that can be executed in Microsoft OS environment. Jar file is container of Java class files, including other resources related to the project.

Jar file can be executed only if java run time environment JRE is present

Jar file format enables you to bundle multiple files into single archive file

Q.15 How is C platform dependent language?

C is platform dependent language since the C compiler is designed to produce platform-specific, optimized code. In C, machine code is different for different processor architecture and thus could not run natively on incompatible platforms unless you have an emulator to help you out.

Q.16 What is the difference b/w path & classpath?

Path is an environment variable that is used to find and locate binary files like "java" and "javac" and to locate needed executable from the command line.

Classpath is an environment variable that is used by the application classloader or system to locate and load the compiled java bytecodes stored in the .class files.