Page No:
Date: / /

Assignment - 2

Name :- Rahul Mankar - KH

Date of Submission :-

Q. 1 what is the difference between JDK, JRE, JVM?

Ans.

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

Q.2 What is JIT compiler ?

Ans. JIT is a part of JVM that optimizes the performance of the application. JIT stands for Java-In-Time Compiler. The JIT is an integral part of the JVM. It accelerates execution performance many times over the previous level. In other words, it is a long-running computer-intensive program that provides the best performance environment. It optimizes the performance of the Java application at compile or run time.

Q.3 What is class loader ?

Ans. The Java classloader is part of JRE that dynamically loads Java classes into the JVM. The Java run time system does not need to know about files and file systems because of classloaders.

A Java classloader is of three types :

1. BootStrap Classloader
2. Extension Classloader
3. System Classloader

Q 4 Explain various memory logical partitions.

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

There are two Memory Management Techniques: Contiguous and Non-Contiguous. In Contiguous Technique, executing process must be loaded entirely in the main memory. Contiguous Technique can be divided into:

1. Fixed (or static) partitioning
2. Variable (or dynamic) partitioning

Fixed partitioning: This the oldest and simple 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 configure.

Q.5 What gives Java its "write once and run anywhere nature"?

Ans. Java language its "write once and run anywhere" nature from its bytecode. Java programs are first converted into class file (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 through. This JAVA intermediate code can be run on any platform be it Windows, Linux provided the system has JRE and JVM installed to run the byte code.

Q.6 Explain History of Java? who invented class?

Ans. 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++ ~~this~~ was an enhancement to the C language, was C languages with classes.

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

Ans. The Java language was initially called "Oak" but was renamed in 1995. Oak was renamed as "Java" because it was already a trademark by Oak Technologies.

Q.8 List features of Java.

Ans. Below are some important features of Java:

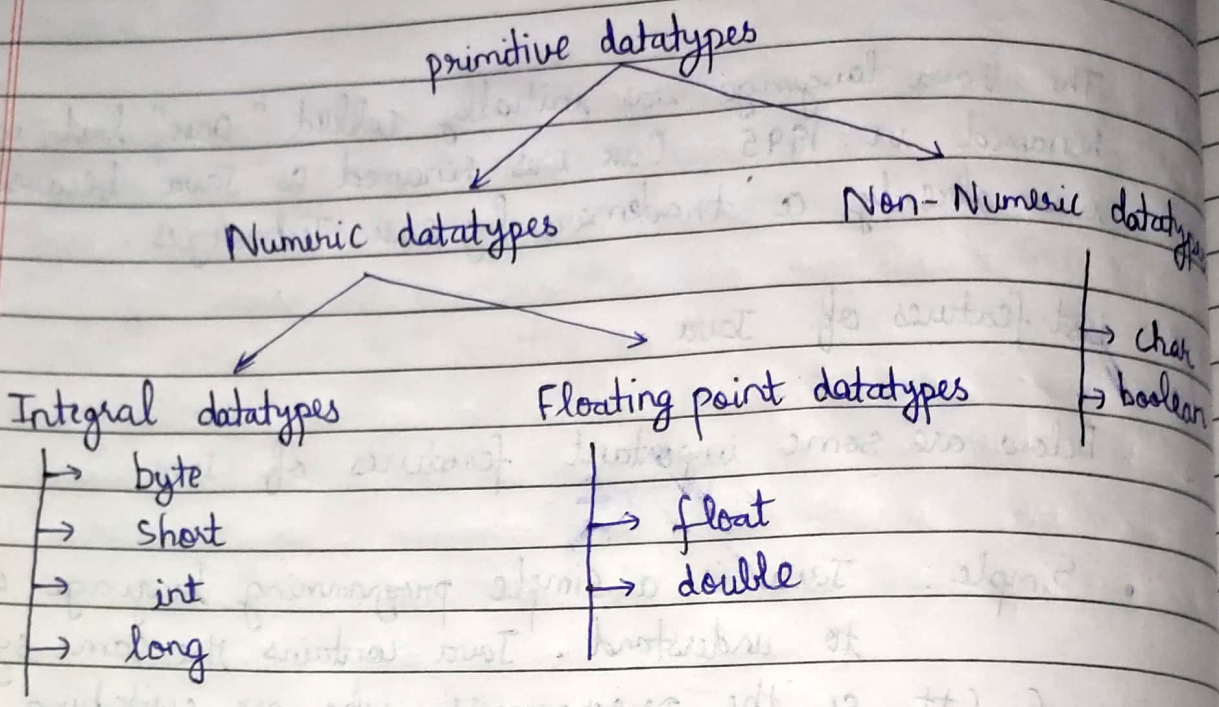
- Simple: Java is a simple programming language and easy to understand. Java contains the same syntax as in C, C++, so the programmers who are switching to Java will not face any problem in terms of syntax. Secondly, the concept of pointers have been completely removed from Java which leads to confusion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

- Objected-Oriented: Java is an Object Oriented Programming Language, which means in Java everything is written in terms of classes and objects.

- Platform Independent
- Portable
- Robust
- Secure
- Interpreted
- Multithreaded

Q.9 List various Datatypes in Java.

Ans.



Q.10 What is the difference between
System.out.print
System.out.println
System.err.print

Ans. System.out refers to the standard output stream
System.err refers to the standard error stream, which also is console by default.

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

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

Page No:
Date: / /

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.?

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

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

Ans. Bytecode is a 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 bytecode.

This may comes 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 between Jar file & Runnable jar file.

Ans. A JAR (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 distribute 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 a 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

Page No:
Date: / /

Q.14 What is difference between Runnable jar file and exe file.

Ans. 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.

The JAR file format enables you to bundle multiple files into single archive file.

Q.15 How is C platform dependent languages ?

Ans. 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 difference between path & classpath ?

Ans. 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 file.