

# Assignment.

i) What is difference between JDK, JRE and JVM?

Ans :-

i) JDK :

- i) JDK stands for Java Development Kit.
- ii) The Java Development Kit is a software development environment which is used to develop Java applications and Applets. It contains JRE and development tools.
- iii) It is often called as superset of JRE.
- iv) JDK contains all the tools required to compile, debug and run the program developed by using Java platform.

JVM  
(Execution Engine)

+ library  
sets + Development  
tools.

JRE

JDK

## Q) JRE

- JRE stands for Java Runtime Environment.
- It is a set of tools responsible for execution of java program on application.
- It is used to provide runtime environment.
- JRE is composed of variety of other supporting software tools and features to get the most out of java application.

JVM  
(Execution Engine)

+ library  
sets

JRE

3) JVM

- JVM stands for Java Virtual Machine.
- JVM loads, verifies and executes source bytecode.
- It is known as 'Interpretation'.
- It is specially responsible for converting bytecode to machine specific code and is necessary in both FOK and JRE.
- It is also platform independent.

2) What is JIT Compiler.

Ans:

JIT stands for Just in time compiler.

- Just in time compiler is a component of the runtime that improves the performance of java applications by compiling.

- JIT is an integral part of JVM
- It is long running, computer intensive program that provides best environment performance.
- It optimizes the performance of the java application at compile or run time.

3) What is class loader?

Ans:-

- The java class loader, is the java runtime environment that dynamically loads java classes into Java virtual machine.
- Java class loader is an abstract class.
- It belongs to java.lang package.
- It is used to load the classes at runtime.
- Class loader's are responsible for loading java classes dynamically to the JVM (Java virtual machine)

5) What gives Java its "Write once and run anywhere nature"

Ans:

Java gets its flora nature from its bytecode.

- Java codes on programs in high level user-friendly language and they are converted into class file (also known as bytecode).
- This means that same Java code can be run on any platform.
- This means programmers can develop Java code on one system and can expect it to run on any other java enabled system without any adjustment.
- Eg, java the program is not connected to co'

Q) Explain History of Java and Who invented Java?

Ans)

The history of Java starts with Java team members

- The principle for creating Java programming language is simple, robust, portable platform-independent, secured, high performance etc.
- Java was developed by James Gosling, who is known as the father of Java in 1995.
- Initially it was designed for small, embedded systems in electronic appliances like set-top boxes.
- Firstly, it was called 'GreenTalk' by James Gosling after that 'Oak' name is given to the language.
- In 1995, Oak was renamed as "Java".

7) What was original name of Java?  
Why it was renamed?

Ans:-

- Firstly, Java is called as "Green talk".
- After that it was called "Oak" and was developed as a part of Green project.
- In 1995, Oak was renamed as "Java" because it was already a trademark by Oak technologies.
- They wanted something that reflects the essence of the technology : revolutionary, dynamic, lively, cool, unique and easy to spell and fun to say.
- Java name was chosen by James Gosling while having a cup of coffee.

## 8) List Features of Java

Apt:

- Simple
- Object-oriented
- Portable
- Platform-Independent
- Secured
- Robust
- Architecture neutral
- Anticipated
- High performance
- Dynamic

## 9) What is difference between

- `System.out.println()`
- `System.out.println()`
- `System.out.println()`

Apt:

- `System.out.println()`

The control moves on the same memory  
on the same line after printing.

- `System.out.println()`

The control moves  
to the next line after printing.

- `System.out.println()`
- 1) `System.out.println()` is used to display common messages.

- 2) The output is displayed in 'red' colour.

- 10) How is Java platform independent?

Ans:-

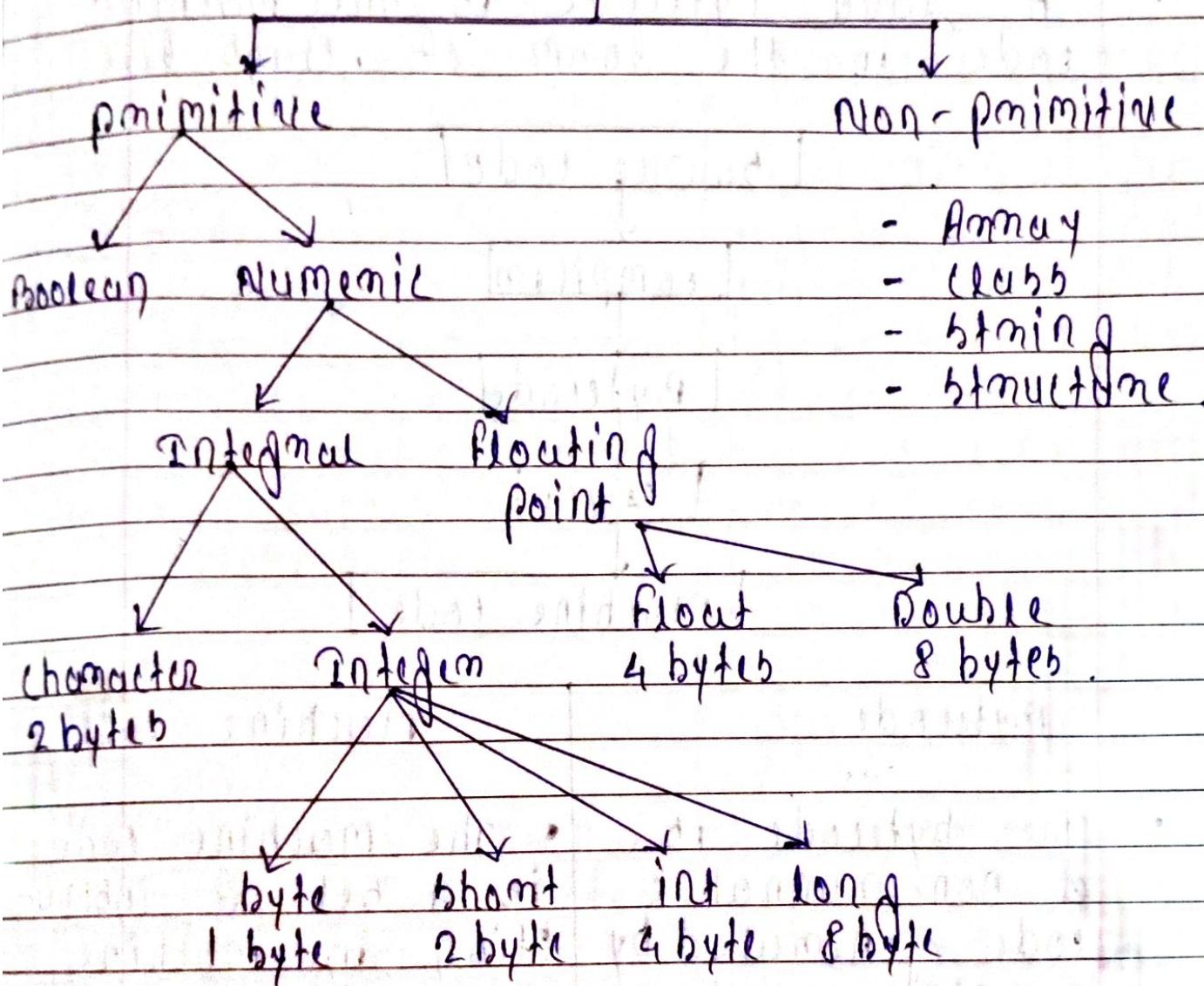
Java is called platform independent because programs written in Java can be run on multiple platforms without re-writing them individually from a particular platform.

- It supports Java bytecode platform independence using Java bytecode and Java virtual machine.
- Java compiler takes contents of program code into byte code. This byte code is platform independent and can run on any JVM operating system.
- JVM interprets the byte code to machine code, and the program is executed.

11) List numerous Datatypes in Java.

Ans)

### Data Types



12) What is Bytecode? How it is different from Machine code?

Ans) Bytecode.

- A bytecode in java is the reason java is platform independent, as soon

as Java program is compiled bytecode is generated.

- A Java bytecode is the machine code in the form of .class file.

source code

compilation

Bytecode

JVM

Machine code

Bytecode

Machine code

- The bytecode is a non-executable code generated by compiling a source code that relies on an interpreter to get executed.

- The machine code is a set of instructions in machine language or binary which can be directly executed by the CPU.

Q) What is the difference between runnable jar file and Runnable jar file.

Ans:-

Jar file

Runnable jar file.

- |   |  |
|---|--|
| 1) Jar file is a Java application which requires a command line to run, a runnable jar file can be directly executed by double clicking.  | 1) Runnable jar file use to run Java classes without having to know class names and type them in command prompt, rather the user can just double click on the jar file and program will fine up. |
| 2) A JAR is a packed file format typically used to aggregate many Java class files associated metadata and resources into one file to distribute application software and libraries on the Java platform. | 2) A runnable jar allows Java classes to be loaded just like when a user clicks on one file.   |

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

Ans:

Runnable .jar file      .exe file.

- 1) Tom file one like dead body.
  - 1)\_exe file one like living men
  - 2) Tom file is combination of compiled files and also contains classes.
  - 2) Executable file is also combination of compiled java classes with main class.

15) How is the platform dependent language?

Ans.)

- C is portable programming language because it is not tied to any hardware or system.
  - We can say, it is a hardware independent language or platform independent language.

c program does not depend on actually but the executable file that is generated at the end from running the c program many depend on platform.

Q) What is difference between Path and Class file?

A)

Path file

Class file.

1) Path is an environment variable which is used by the operating system to find the compiler to the executables.  
2) Classpath is nothing but setting up the environment for Java compiler to find the path of classes.

2) Path is nothing but setting up an environment for executables.

3) In Path we set the path of executables.  
3) In Classpath we set path of jars from compiling classes.