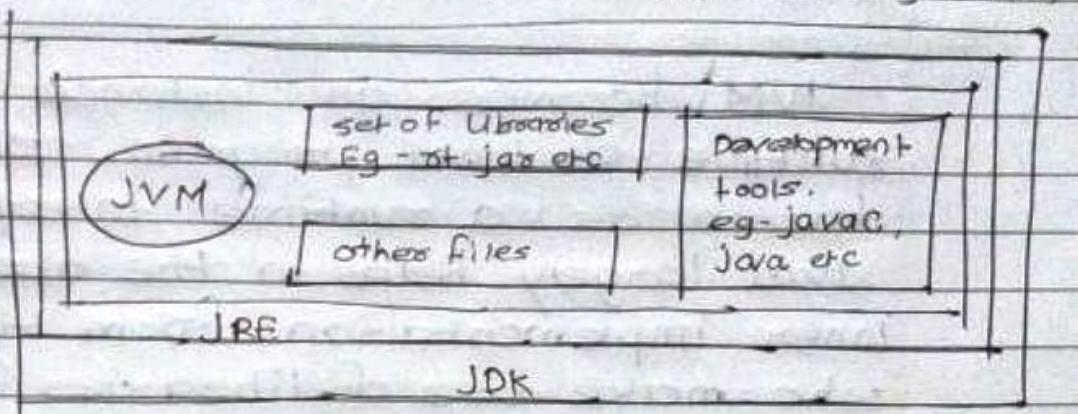


## Assignment - 2



Q-1 What is difference between JDK, JRE & JVM.



i) **JDK (Java Development Kit)** - is a kit that provides the environment to develop and execute (run) the java program. JDK is a kit (or package) that includes two things.

ii) **Development Tools** (to provide an environment to develop your java programs)  
ii) **JRE** (to execute your java program).

iii) **JRE (Java Runtime Environment)** is an installation package that provides an environment to only run (not develop) the java program (or application) onto your machine. JRE is only used by those who only want to run Java programs that are end-users of your system.

iii) **JVM (Java Virtual Machine)** - is a very important part of both JDK & JRE because it is contained as inbuilt in both. whatever Java program you run using JRE or JDK goes into JVM & JVM is responsible

executing the Java program line by line, hence it is also known as an interpreter.

JVM becomes an instance of JRE at the runtime of Java program. It is widely known as a runtime interpreter.

JVM largely helps in the abstraction of inner implementation from the programmers who make use of libraries for their programs from JDK. It is mainly responsible for three activities

- i) Loading
- ii) Linking
- iii) Initialization.

## Q-2] What is JIT Compiler?

- i) The Just-In-Time(JIT) Compiler is an essential part of the JRE i.e - Java Runtime Environment, that is responsible for performance optimization of java based applications at run time.
- ii) In order to improve performance, JIT Compilers interact with the Java Virtual Machine (JVM) at run time and compile suitable bytecode sequences into native machine code;
- iii) While using a JIT compiler, the hardware is able to execute the native code, as compared to having the JVM interpre-

DATE \_\_\_\_\_

the same sequence of bytecode repeatedly and incurring overhead for the translation process. This subsequently leads to performance gains in the execution speed, unless the compiled methods are executed less frequently.

a-3) What is class loader?

- ii) The Java Class Loader is a part of the Java Runtime Environment that dynamically loads Java class into the Java Virtual Machine. The Java runtime system does not need to know about files & file systems because of classloaders.
- iii) Java classes aren't loaded into memory all at once, but when required by an application. At this point, the Java class loader is called by the JRE and the class loader loads classes into memory dynamically.

Depending on the type of class and the path of class, the classloader that loads that particular class is decided. To know the classloader that loads a class the `getClassLoaders()` method is used. All classes are loaded based on their names and if any of these classes are not found then it returns a `NoClassDefFoundError` or `ClassNotFoundException`.

A java classloader is of three types.

- 1) **Bootstrap classLoader** - A bootstrap classloader is a Machine code which kickstarts the operation when the JVM calls it. It is not a java class. Its job is to load the first pure java classLoader. Bootstrap classLoader loads classes from the location rt.jar. Bootstrap classLoader doesn't have any parent classLoader. It is also called as the primordial classLoader.
- 2) **Extension classLoader** - The Extension classLoader is a child of Bootstrap classLoader and loads the extensions of core java classes from the respective JDK Extension library. It loads files from jre/lib/ext directory or any other directory pointed by the System property java.ext.dirs.
- 3) **System Class Loader** - An Application classLoader is also known as a System classLoader. It loads the Application type classes found in the environment variable, CLASSPATH, -classpath or -cp Command line option. The Application classLoader is a child class of Extension classLoader.

Q-4) Explain various memory logical partitions.

→ Memory partitioning is the system by which the memory of a computer system is divided into sections for use by the resident programs. There are different ways in which memory can be partitioned: Fixed, Variable & dynamic partitioning.

i) Fixed partitioning - In this technique, the main memory is divided into partitions of equal or different sizes. The operating system always resides in the first partition while the other partitions can be used to store user processes. The memory is assigned to the processes in contiguous way. In fixed partition,

- ii) The partitions can not overlap
- iii) A process must be contiguously present in a partition for the execution.

ii) Variable partitioning - Variable partitioning is a part of the contiguous allocation technique. It is used to alleviate the problem faced by fixed partitioning. As opposed to fixed partitioning, in variable partitioning, partitions are not created until a process executes. Initially RAM is empty & partitions are made during the run-time according to process's need instead of

partitioning during system config.

The size of partition will be equal to incoming process.

The partition size varies according the need of the process so that the internal fragmentation can be avoided to ensure efficient utilisation of RAM.

3) Dynamic partitioning - Dynamic partitioning tries to overcome the problems caused by fixed partitioning. In this technique, the partition size is not declared initially. It is declared at the time of process loading.

The first partition is reserved for the OS. The remaining space is divided into parts. The size of each partition will be equal to the size of the process. The partition size varies according to the need of the process so that the internal fragmentation can be avoided.

### Advantages

- 1) No internal fragmentation
- 2) No limitation on the size of the process
- 3) Degree of multiprogramming is dynamic

### Disadvantages -

- 1) External fragmentation
- 2) Complex Memory Allocation.

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

- i) Java applications are called WORA (Write Once RUN Anywhere). This means a programmer can develop Java code on one system and can expect it to run on any other Java-enabled system without any adjustment. This is all possible because of JVM.
- ii) In traditional programming languages like C, C++ when programs were compiled, they used to be converted into the code understood by the particular underlying hardware, so if we try to run the same code at another machine with different hardware, which understands different code will cause an error, so you have to re-compile the code to be understood by the new hardware.
- iii) In Java the program is not converted to code directly understood by hardware rather it is converted to bytecode (.class file), which is interpreted by JVM, so once compiled it generates bytecode file, which can be run anywhere (any machine) which has JVM (Java Virtual Machine) and hence it gets the nature of write once & run anywhere.

a-6) Explain history of Java. Who invented Java?

- i) Java is an object-oriented programming language developed by James Gosling in the early 1990's at Sun Microsystems.
- ii) Initially it was designed for small, embedded systems in electronic appliances like set-top boxes.
- iii) Firstly, it was called "GreenTalk" by James Gosling, and the file extension was .gt. The small team of Sun engineers called Green Team.
- iv) After that, it was called Oak & was developed as a part of the Green project. In 1995, Oak was renamed as "Java" because it was already a trademark by Oak Technologies.
- v) Java is an island in Indonesia where the first coffee was produced called Java coffee. It is a kind of espresso bean. Java name was chosen by James Gosling while having a cup of coffee nearby his office.
- vi) In 1995, Time magazine called Java one of the Ten Best products of 1995.
- vii) Many Java versions have been released now.



a-8) What was original name of Java? Why it was renamed?

- 1) Firstly Java was called "green talk" by James Gosling, developer of Java language.
- 2) After that, it was called Oak & was developed as a part of the green project. In 1995, Oak was renamed as "Java" because it was already a trademark by Oak Technologies.
- 3) Java is an island in Indonesia where the first coffee was produced called Java coffee. It is a kind of espresso bean. Java name was chosen by James Gosling while having a cup of coffee nearby his office.

a-9) List features of Java?

- 1) platform Independent - Compiler Converts source code to bytecode and then JVM executes the bytecode generated by the compiler. This bytecode can run on any platform be it windows, Linux, or mac OS. Which means if we compile a program on windows, then we can run it on Linux & vice versa.

2) object-oriented programming Language - organizing the program in the terms of a collection of objects is a way of object oriented programming. Each

which represents an instance of the class. The four main concepts of oop are -

- 1) Abstraction
- 2) Encapsulation
- 3) Inheritance
- 4) polymorphism.

3) Simple - simple language as it does not have pointers, operator overloading, multiple inheritances, etc.

4) Robust - reliable language, java compiler is able to detect even those errors that are not easy to detect by another programming language.

5) Secure - Java programs run in an environment that is independent of the os environment which makes java programs more secure.

6) Distributed - we can create distributed applications using the java programming language. programs can be easily distributed on one or more systems that are connected to each other through an internet connection.

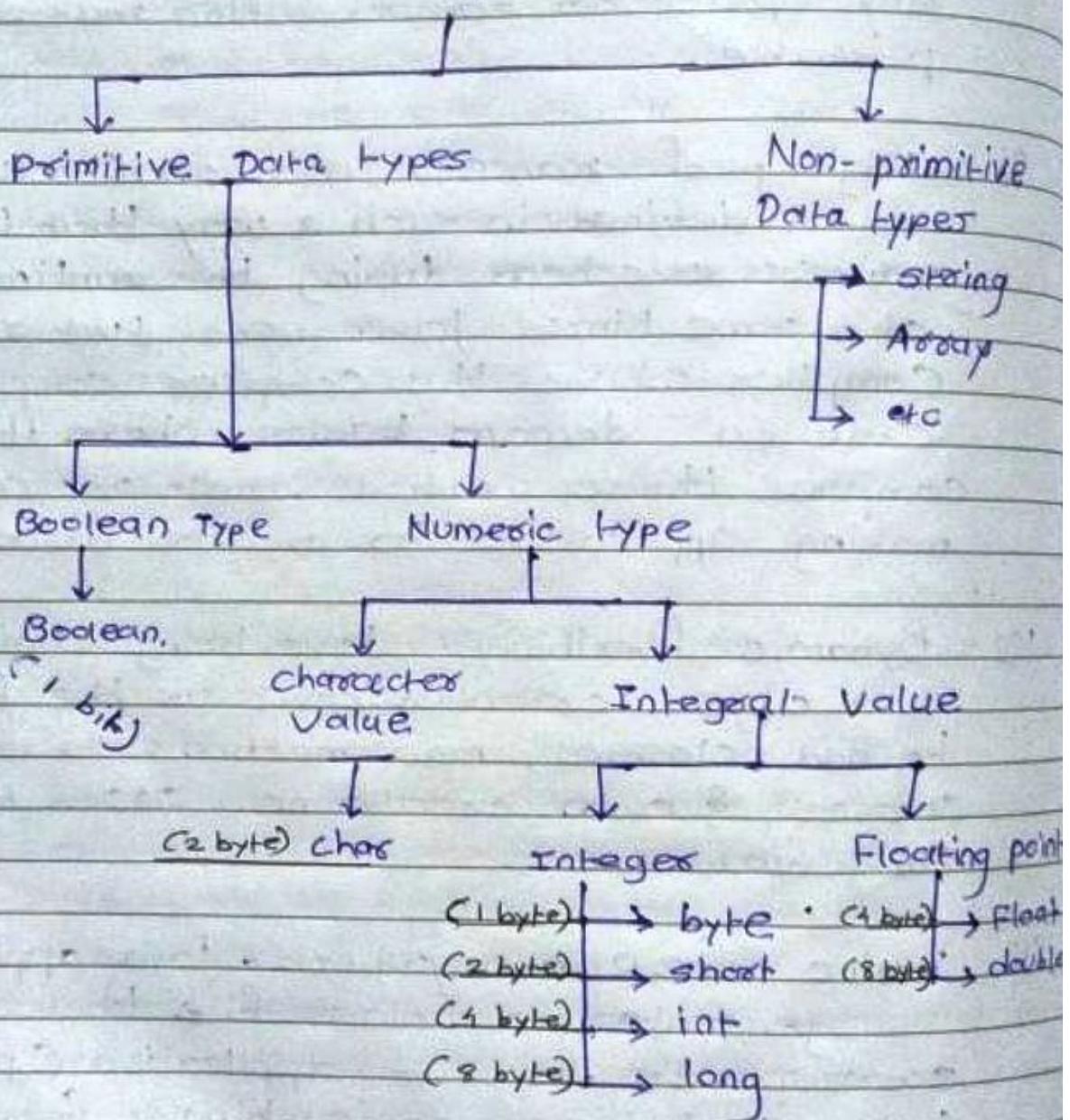
7) Multithreading - It is a java feature that allows concurrent execution of two or more parts of a program for max<sup>m</sup> utilization of the cpu.



- 8) Portable - The platform independent feature of Java in which its platform independent bytecode can be taken to any platform for execution makes java portable.
- 9) High performance - Java architecture is defined in such a way that it reduces overhead during the runtime and at some times Java uses Just In Time Compiler where the compiler compiles code on demand basis where it only compiles those methods that are called making applications to execute faster.
- 10) Dynamic flexibility - Java being completely object oriented gives us the flexibility to add classes, new methods to existing classes, & even create new classes through Sub-classes.
- 11) Write once Run anywhere - Java application generates a class file that corresponds to our application (program) but contains code in binary format.  
It provides easy to architecture nature neutral ease as bytecode is not dependent on any machine code.

e-9) List Various data types in Java.

→ Data types in Java.



e-10) What is difference between?  
System.out.print  
System.out.println  
System.out.printf

→ System.out is a print stream to which we can write characters, it outputs

We write to it on the Command Line Interface console/ terminal. It is mostly used for console applications/ programs to display the result to the user.

i) `System.out.print()` - prints the argument only, without adding any space or new line after the printing is done. The next print statement starts printing immediately adjacent to the end of the last print statement.

ii) `System.out.println()` - prints the argument passed to it and then also prints a new line. The next statement after `println()` is printed in a new line.

iii) `System.err.print()` - print to the standard error. `System.err.println()` is mostly used to output error texts. It also gives output on the console but most of the IDE's gives it a red color to differentiate.

`System.err` & `System.out` both are the same both are defined in `System` class as reference variable of `PrintStream` class.

Q-11) How is Java platform independent?

→ In Java the program is not directly converted to code directly understood by hardware, rather it is converted to bytecode (.class) file which is interpreted by JVM. So once compiled it generates bytecode file, which can be run anywhere, (on any machine) which has JVM. If we compile a program on windows, then we can run it on Linux & vice versa.

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

→ Byte code is an intermediate code between the source code and machine code. It is a low-level code that is the result of the compilation of a source code which is written in high-level language. It is processed by a virtual machine like (Java Virtual Machine) JVM. It is compiled to run on JVM, any system having JVM can run it irrespective of their operating system. Byte code makes Java platform independent. Bytecode is referred as portable code.

Machine Code - machine code is a set

DATE \_\_\_\_\_  
PAGE \_\_\_\_\_

of instructions that is directly machine understandable & it is processed by the central processing unit (CPU).

Machine code is in binary (0's & 1's) format which is completely different from the byte code & source code.

It is regarded as the most lowest representation of the source code.

Machine code is obtained after compilation or interpretation. It is also called machine language.

Q-13) What is difference between jar file & Runnable jar file?

→ With standard JAR file, you have to specify the class with the main method on the command line when running the JAR. With runnable JAR, there is a manifest file that will hold that information so you can just type java/javam myRunnable.jar or simply double click it.

The runnable jars contains a MANIFEST.MF file, which defines the main class to be executed when the jar is run.

Non-runnable jars are just libraries of classes, that can be added to the classpath so that code is reused (it also contains the manifest file, but no main class there).

A runnable jar is a jar file that has an embedded manifest file that includes the "Main-class:" declaration. The "Main-class" must be defined so the java runtime knows which class to call when the jar is "run." If a jar does not include a manifest with the "main-class:" it is not considered a "runnable jar" - it is just a library of Java code.

Q-15) What is the difference bet" 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. The Java™ Archive (JAR) file format enables you to bundle multiple files into a single archive file.

The class files compiled from java files, can not be launched directly. That is why it is needed to be converted to exe before, it can run in a windows environment. The usual way to start a java program by batch file is not a convenient way, so in order to avoid this difficulty we need to convert jar files into exe file.



Also converting it to .exe enables the program to run by simple double click on the program, instead of having to compile it with an IDE or through the JVM.

Q-15]

How is C platform dependent language?

→ In traditional programming languages like C & C++, when programs were compiled, they used to be converted into the code understood by the particular underlying hardware. So if we try to run the same code at another machine with different hardware, which understands different hardware code will cause an error. So you have to re-compile the code for the code to be understood by the new hardware. This is why C is platform dependent language.

Q-16] What is difference between path &amp; classpath?

→ Path - PATH is an environment variable that is used to find & locate binary files like "java" & "javac" & to locate needed executables from the command line or terminal window. To set the path, we're supposed to include or mention JDK-Home/bin directory in a PATH environment variable.

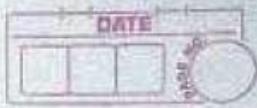
The PATH can not be overridden by providing command and PATH is only used by the operation system(OS) to find binary files.

Classpath - Classpath is an environment variable that is used by the application classloader or system to locate & load the compiled Java bytecodes stored in the .class file. To set CLASSPATH, the CLASSPATH can be overridden by adding classpath in the manifest file & by using a command like set -classpath. The CLASSPATH is only used by Java classloader to load class files.

### PATH

### CLASSPATH

- |   |  |
|---|--|
| 1) An environment variable is used by the OS to find the executable files               | 1) An environment variable is used by the Java compiler to find the path of classes.     |
| 2) PATH setting up an environment for the OS, OS will look in this PATH for executables | 2) Classpath setting up the environment for Java, Java will use to find compiled classes |
| 3) Refers to the operating system,  | 3) Refers to the developing Environment  |



- 4) In path Variable, we must place .\bin folder path
- 4) In classpath, we must place .\lib\jar file or directory path in which .java file is available.
- 5) PATH is used by cmd prompt to find binary files
- 5) CLASSPATH is used by the Compiler & JVM to find library files.

---\*\*\*---