Page No.
Date

Assignment no. 2

1) What is the difference between JDK, JRE and JVM.

→ -Java Development Kit (JDK) is a software development environment used for developing Java applications & applets. It includes the Java Runtime Environment (JRE)

-JRE is an acronym for Java Runtime Enviroment. It is also written as Java RTE. The Java Runtime Enviroment is a set of software tools which are used for developing Java applications. It is used to provide the runtime environment. It is the implementation of JVM. It physically exists. It contains a set of libraries + other files that JVM uses at runtime.

- JVM (Java Virtual Machine) is an abstract machine. It is called a virtual machine because it doesn't physically exist. It is a specification that provides a runtime enviroment in which Java bytecode can be executed. It can also run those programs which are written in other languages and complied to Java bytecode.

- JVMs are available for many hardware & software platforms. JVM, JRE & JDK are platform dependent because the config. of each OS is different from each other. However, Java is platfoem independent.

2) What is JIT Compiler?

→ The Just-In-Time (JIT) compiler is a component of runtime environment that improves the performance of Java applications by compiling bytecodes to native machine code at run time.

- Advantages
- It requires less memory usages
- The code optimization is done at run time.
- It uses different levels of optimization.

- It reduces the Page faults.

3) What is class loader?

→ The Java class Loader is a part of the Java Runtime Environment that dynamically loads Java classes into the Java virtual Machine. The Java run time System does not need to know about files & file Systems because of class loaders. 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 & these class Loaders load classes into memory dynamically.

- Types
1. Boot strap class Loader
2. Extension class Loader
3. System class Loader.

4) Explain Various memory Logical partitions.

→ -A Logical partition (LPAR) is the division of a computer's memory, and storage into multiple sets of resources so that each set of resources can be operated independently with its own operating system instance and applications. The number of logical partitions are used for different purposes such as data base operative or client/server operation or to separate test and production environments.
- Each partitions can communicate with the other partitions as if other partition is in a separate machine.
-

- Logical partitioning was first studied by IBM in 1976 and later introduced by Amdahl and then IBM. Hitachi and Sun Microsystems also use forms of logical partioning. Today both IBM's S/390 and AS/400 products support logical partitioning.

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

⇒ • One of the initial "killer feature" of Java was supposed to be the write once, run anywhere nature of it. Earlier, it is not practically possible to have different versions of an application for different devices because the devices have variety of CPU's, operating system & browsers. The same code must work on all the computers, therefore we need a portable code. Portability refers to the ability to run a program on different machines. "Java is portable", means that you can run Java bytecode on any hardware that has a compliant JVM (Java Virtual Machine).

• The Java compiler compiles a Java program (.java file) and converts it into class files (.class) that contains bytecodes, which is the intermediate language between source code & machine code. These bytecodes are not platform specific, so with the help of JVM, the java progeam can run on wide variety of platforms.

6) Explain History of Java. Who invented Java?

→ - Java was originally designed for interactive television but it was too advanced technology for the digital cable television industry at the time. The history of Java starts with the green Team. Java team members (also known as Green Team), initiated this project to develop a language for digital devices such as set-top boxes, televisions, etc. However, it was best suited for internet programmi Later, Java technology was incorporated by Netsca
- Java was developed by James Gosling, who is known as the father of Java in 1995. James Gosling & his team members started the project in the early '90s.

1) James Gosling, Mike Sheridan, & Patrick Naughton initiated the Java language project project in June 1991. The small team of Sun engineers called Green Team.

2) Initially it was designed for small, embedded systems in electronic appliances like set-top boxes

3) Firstly, it was called "Greentalk" by James Gosling, & the file extension was .gt.

4) After that, it was called OAK & was developed as a part of the green project.

5) In 1995, Oak was renamed as "Java" because it was already a trademark by Oak Technologies

7) Java is an island in Indonesia where the first coffee was produced. Java name was chosen by James Gosling while having a cup of coffee nearby his office.

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

→ - Firstly, it was called "Green talk" by James Gosling, & the file extension was .gt.
- 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.

8) List features of Java.

→ The primary objective of Java programming language creation was to make it portable, simple & secure programming language. Apart from this, there are also some excellent features which play an important role in the popularity of this language. The features of Java are also known as Java buzzwords. A list of most imp. features of the Java language is:

1. Simple
2. Object-Oriented
3. Portable
4. Platform independent
5. Secured
6. Robust
7. Architecture neutral
8. Interpreted
9. High Performance
10. Multithreaded
11. Distributed
12. Dynamic

9) List various Datatypes in Java.

→ There are two types of data types in Java:-

1. Primitive :
   boolean, char, byte, short, int, long, float & double.
2. Non-primitive data types :
   Classes, Interfaces, Arrays.

Data type	Default Value	Default Size
boolean	false	1 byte
char	\u0000	2 byte
byte	0	1 byte
short	0	2 byte
int	0	4 byte
long	0L	8 byte
float	0.0f	4 byte
double	0.0d	8 byte.

11) How is Java platform independent?

→ Java provides platform independence by making use of Java byte code. Java Byte code or .class file is generated during the compilation of the code. This Byte code is platform-independent & can run on any system regardless of the platform it is built upon.

10) What is difference btn System.out.print, System.out.println & System.err.print.

⇒ - System.out.print() prints text in a line. Consecutive runs of the same command print text on the same line. If you want to move to the next line, you have to append a line break to the arguments of the command. The cursor remains on the same line after printing the text. The print method only works with arguments otherwise it will cause a syntax error.

- System.out.println() prints a line of text, and moves to the next line automatically, no need to append a line break. If you add a line break, it's gonna make a new line, Kinda like a Paragraph Style text. The cursor moves to the next line after printing the text. The println method can work without arguments.

- System.out is "standard output" (stdout) and System.err is "error output" (stderr). System.err.println() will print to standard error This stream is already open and ready to accept output data.

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 a high-level language. It is processed by a virtual machine like Java virtual Machine (JVM). Byte code is a non-runnable code after it is translated by an interpreter into machine code then it is understandable by the machine. It is compiled to run on JVM.

Byte code	Machine code
- Byte code consisting of binary, hexadecimal, macro instructions & it is not directly understandable by the CPU.	- Machine code consisting of binary instructions that are directly understandable by the CPU.
- Byte code is considered as the intermediate-level code	- Machine code is considered as the low-level code.
- Byte code is a non-runnable code generated after compilation of source code and it relies on an interpreter to get executed.	- Machine code is a set of instructions in machine language or in binary format & it is directly executed by CPU.
- Byte code is executed by the virtual machine then the Central Processing Unit.	- Machine code is not executed by a virtual machine it is directly executed by CPU

Page No.
Date

13) What is difference between Jar file & Runnable Jar file

→ - In simple terms, the difference between JAR file and Runnable JAR is that while a JAR file is a Java application which requires a command line to run, a runnable JAR file can be directly executed by double clicking it.

- A JAR (Java Archive) is a package file format typically used to aggregate many Java class files & associated metadata and resources into one file to distribute application software or libraries on the Java platform.

- In simple words, a JAR file is a file that contains a compressed version of .class files, audio files, image files, or directories. We can imagine a .jar file as a zipped file (.zip) that is created by using WinZip Software. Even, WinZip Software can be used to extract the contents of a .jar.

- A runnable jar file allows a use to run Java classes without having to know class names and type them in a command prompt, rather the user can just double click on the jar file and the program will fire up. A runnable jar allows Java classes to be loaded just like when a user clicks an exe file.

14) What is difference between Runnable Jar file & exe file.

→ Jar file are like dead body, exe file are like living men. Jar file is the combination of compiled java classes. Executable Jar file is also combination of compiled java classes with main class.

15) How is C platform dependent language?

→ - C is a 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. That is why C is called portable language.

- C programs does not depend on platforms actually. But, the executable file that is generated at the end for running the C-program may depend on a platform.

- When you use OS you get other extension for executable files. Example, When we use Mac for programming in C, when we compile it, & run it. "out" is generated. This file is generated when you compile it using the terminal of macOS or any other Unix terminal.

16) What is difference between path & classpath.

→ - The main difference between PATH & CLASS PATH is that Path is set for java tools in java programs like java and javac, which are used compile your code. Whereas classpath is used by system or Application class loader to locate and load compile Java bytecodes stored in the .class file.

- The classpath is a parameter in the Java Virtual Machine (JVM) or the JAVA compiler that is used by a system or application class loader to locate and load compiled that is also an environment variable path that behaves as a mediator between the operating system and developer to inform binary file path.

- Path is an environment variable that is used to find and locate binary files like "java" and "javac" and 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 variables. The Path can not be overridden by providing command & Path is only used by the operation system (OS) to find binary files.

H.W. (Day 7)

1) Identify the features for the following versions :- Java 8, 15, 17, 19.

→ 1) Java 8

- Oracle released a new version of Java as Java 8 in March 18, 2014. It includes various upgrades to the Java programming, JVM, Tools and Libraries.
- Java 8 provides following features :-
  - Lamda expressions
  - Method references.
  - Functional interfaces
  - Stream API
  - Default methods
  - Base64 Encode Decode
  - Static methods in interface.
  - Optional class
  - ForEach() method
  - Nashorn JavaScript Engine
  - Parallel Array Sorting
  - Type and Repating Annotations
  - Concurrency Enhancements
  - JDBC Enhancements