Hrishikesh Dhamare - KH

Date / /

Page L&W

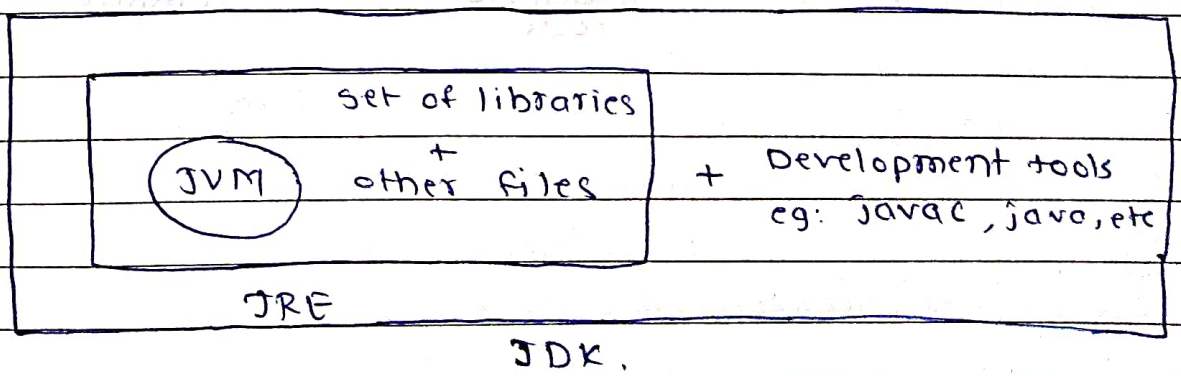
Assignment - 02.

Q1. Difference between JDK, JRE and JVM.

Ans: ① Java Development kit (JDK) is a software development environment used for developing Java application and applets. It includes the Java Runtime Environment (JRE), an interpreter/loader, a compiler, an archiver (jar), a documentation generator (Javadoc), and other tools needed in Java development.

② The Java Runtime Environment (JRE), provides the minimum requirement for executing a Java application; it consists of Java Virtual Machine (JVM) core classes, and supporting files.

③ Java Virtual Machine (JVM) is very important part of both JDK and JRE because it is contained or inbuilt in both. JVM is responsible for executing the java program line by line, hence also called as interpreter.



Q2. What is JIT compiler ?

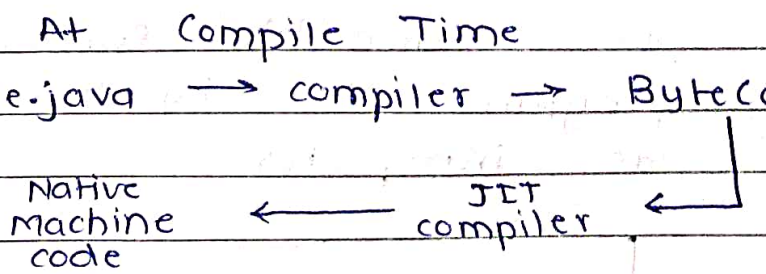
Ans

① Just-in-Time (JIT) compiler is essential part of JRE, that is responsible for performance optimization of java based application at run time.

② JIT compiler compiles bytecode to native machine code at run time.

③ Interpretting the bytecode affects the speed of execution in order to improve performance, JIT compiler interact with JVM at run time and compile suitable bytecode sequence into native machine code.

④ Optimization performed by JIT compiler are data analysis, reduction of memory accesses by register allocation, translation from stack operations to register operations, etc.



Date / /
Page L&W

Q3. What is class loader ?

Ans. ① The Java class loader is part of the Java Runtime Environment that dynamically loads java class into JVM.

② The Java runtime system does not need to know about files and file systems because of classloaders.

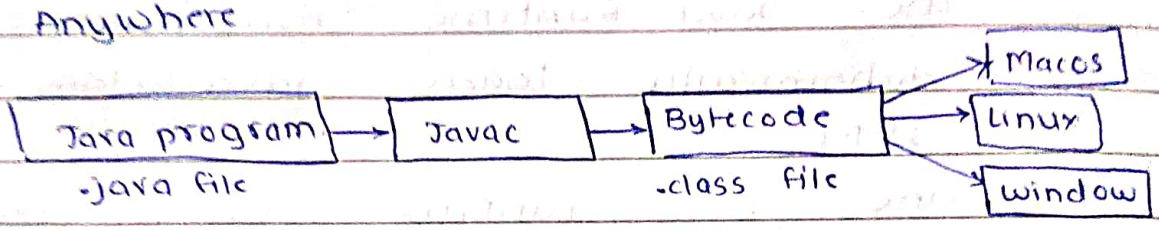
③ Java classes aren't loaded into memory all at once, but when required by an application. At this point, the Java class loader is called by the JRE and these classloader load classes into memory dynamically.

Q4. What gives Java its "write once and run anywhere nature"?

Ans: ① Java applications are called WORA (write once and run anywhere) This means a programmer can develop Java code on one system and can expect it to run on any other Java-enable system without any adjustment. This is all posible because of JVM.

② 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 and hence it gets the nature of write once and Run Anywhere



Q5. Explain History of java who invented Java?

Ans
- The principle for creating Java programming were " Simple Robust, Portable, Platform - independent, Secured High performance, multi-threaded, object-oriented interpreted."
- Java was launch in 1991, by sun Microsystem USA.
- Java was developed by James Gosling.
- Initially the 'java was given name OAK
- In 1995, Oak was renamed as 'Java'

Q6. What was original name of Java ? why it was renamed ?

Ans
- Firstly it was called "Greentalk" by James Gosling, and the file extensions was .gt.
- After that, it was called "Oak" and was developed as part of Green project
- In 1995, Oak was renamed as "Java" because it was already a trademark by Oak Technologies.

Q7. list the features of JAVA.

Ans
1. Simple
2. Robust
3. Object oriented
4. Platform independent
5. Secured
6. ~~Robut~~ Architecture Neutral
7. Interpreted
8. High Performance
9. Multi-threaded
10. Distributed
11. Dynamic
12. Portable.

Q8. List various Datatypes in JAVA

Ans
1. Primitive data type : The primitives data types include boolean, char, byte, short, int, long, float and double.
2. Non-primitive data types : The non-primitive data type include interface, and array.

Q9. Difference between System.out.print, System.err.print and System.out.println

Ans.
1. System.out.print only print the content without switching to next line.
2. System.out.println print the ~~contist~~ content and switch to the next line after execution.
3. System.err.print will print to the standard error.

Date / /
Page L&W

Q10. How is Java platform independent?

Ans. The meaning of platform - independent is that the java compiled code (bytecode) can run all operating systems.

- The compiler converts the high - level language into a format understand by machines.
- Therefore, a compiler is a program that translates the source code for another program from a programming language into executable code.
- The result of JAVA compiler is the .class file and not machine language.
- This bytecodes are converted by JVM into native machine lang.
- The JVM take care of difference the bytecode for different platform which makes them platform independent.

Q11. What is bytecode? How it is different from machine code?

Ans. The bytecode is created by compiler by converting source code into bytecode. The bytecode generated is a non-executable code and need intrepreter to execute on a machine. This interpreter is JVM and thus the Bytecode is executed by JVM.

Date / /
Page L&W

Q12. What is difference between Runnable jar file & exe file?

Ans. 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 user clicks an .exe file.

- An executable file (exe file) is a computer file that contain an encoded sequence of instruction that the system can execute directly when the user clicks the file icon.

Q13. What is Jar file?

Ans: Jar stands for Java Archive. jar files are packaged with the ZIP file format so you can use them for tasks such as lossles data compression, archiving, decomposition, and archive up unpacking.
These tasks are among the most common uses of JAR files, and you can realize many JAR files benefits using only these basic features.

Date / /
Page L&W

Q14. How is C platform dependent language?

Ans It is platform dependent because the C compiler generates a machine code which can be understand by the respective platform. So, Obj file generated is different for different platforms. so it became a platform dependent.

Q15. What is difference between path and class path?

Ans ① Path is an environment variable that is used to find and locate binary files like "java" and "javac" and to allocate needed executable from command line or terminal window.

② The class path is parameter in the Java virtual machine (JVM) that is used by system or application class loader to locate and load compiled Java bytecodes stored in the ".class" files.