**Assignment 3**

1. Explain the components of the JDK.

Java Development Kit

It contains Java Development Tools, Java APIs, Java Supporting libraries, and the JRE Java execution/Runtime environment and the JVM(Java Virtual Machine) for the development of Java based Software application.

Src.zip – contains Source code of Java APIs - .java files

Rt.jar – contains compiled code of Java APIs - .class files

Documentation of Java APIs

1. Differentiate between JDK, JVM, and JRE.

JDK :

JDK stands for Java Development Kit.

It is a platform dependent i.e separate for Windows, Mac, Linux.

It contains Java Development tools like compiler, interpreter, java API docs, JRE

JDK Components are : Java Development Tools, Java APIs Documentation, Java API libraries(rt.jar), JVM, and Src.zip which contains Source code samples.

JRE :

JRE stands for Java Runtime Environment

It is also platform dependent.

It consists of JVM, Java libraries and other classes for smooth execution of programs

To run or execute the java application on Developer’s machine or the Client’s machine we require the JRE

Components of JRE are : Rt.jar file – Java class library and JVM – Java Virtual Machine

JVM :

JVM stands for Java Virtual Machine

It is responsible for executing the java program line by line so can be called as interpreter

With the help of JVM we can run the class file on any of the machine.

1. What is the role of the JVM in Java? & How does the JVM execute Java code?

JVM Java Virtual Machine runs the Java applications as a run time engine.

JVM is the one that calls the main method present in java code.

When we compile a .java file , a .class file the byte code with the same class names present in .java file are generated.

The program written on one system the class file can be executed on any of the machine with the help of JVM – so it is said to be as Write Once and Run Anywhere.

1. Explain the memory management system of the JVM.

JVM acts as a run-time engine to run Java applications. JVM is the one that actually calls the main method present in java code. JVM is the part of JRE.

The memory in the JVM is divided into 5 different parts:

1. Method Area :

Method area is the memory block that stores the class code, variable code, (static variable, runtime constant)

1. Heap Area :

Objects are stored in the Heap area.

1. Java Stack :

It creates separate stack for each thread, all method local variables are stored here.

1. PC Registers :

It stores currently executing instructions

1. Native Method Area :

When the threads are created are stored in the Native Method Area.

1. What are the JIT compiler and its role in the JVM? What is the bytecode and why is it important for Java?

JIT (Just in Time) compiler is the part of JRE responsible for performance optimization.

JIT Compiler improves the performance of Java code by compiling the java code into Native Machine Language i.e. Hexadecimal codes at run time. It stores the Native Machine Codes in cache memory for faster execution

1. Describe the architecture of the JVM.

1)Class Loader Subsystem:

a. Bootstrap Class Loader

b. Extension Class Loader

c. System Class Loader

d. Custom Class Loader

2)Runtime Data Areas

a. Method Area

b. Heap Area

c. Java Stack

d. PC Registers

e. Native Method Stack

3) Execution Engine

a. Interpreter

b. JIT Compiler

c. Garbage Collector

1. How does Java achieve platform independence through the JVM?

When the Java program is compiled, the .class file the Bytecode is get generated.

This Bytecode .class file is executable on any machine with the help of JVM

So likewise the OS dependent JVM makes the Java platform Independent.

Write once and Run Anywhere.

1. What is the significance of the class loader in Java? What is the process of garbage collection in Java.?

Class Loader is the part of JRE which helps to load the java classes into JVM and dynamically loads the classes in memory as needed.

Garbage Collection in Java automatically performs the memory management. If any unused objects, it clears the memory.

9)What are the four access modifiers in Java, and how do they differ from each other?

1. private – with private, the members of the class are accessible only within the class.

2. protected – protected data members are accessible within the same package and the subclass of different package

3. public – The public data members are accessible everywhere.

4. Package Level private (default) – the default members are accessible within the package.

10) What is the difference between public, protected, and default access modifiers?

1. private – with private, the members of the class are accessible only within the class.

2. protected – protected data members are accessible within the same package and the subclass of different package

3. public – The public data members are accessible everywhere.

4. Package Level private (default) – the default members are accessible within the package.

11) Can you override a method with a different access modifier in a subclass? For example, can a protected method in a superclass be overridden with a private method in a subclass? Explain.

To override the method we need to keep the access modifiers same for all overridden methods.

We cannot override protected method with private method.

12) What is the difference between protected and default (package-private) access?

protected – protected data members are accessible within the same package and the subclass of different package

Package Level private (default) – the default members are accessible within the package.

Default members are not accessible at all in different packages.

13) Is it possible to make a class private in Java? If yes, where can it be done, and what are the limitations?

Yes, you can declare the a class as private in nested class. First class should be public and in that private class can be created.

With this the private class will be accessible within the enclosing class. It can be used for encapsulation.

14) Can a top-level class in Java be declared as protected or private? Why or why not?

The top-level class in Java cannot be declared as the protected or private as it will not be accessible from other classes within the same package as well.

15) What happens if you declare a variable or method as private in a class and try to access it from another class within the same package?

If we declare the variable or method as private and try to access it from the another class within same package, it won’t be possible to access the private members from different classes. The compilation error will occur.

16) Explain the concept of "package-private" or "default" access. How does it affect the visibility of class members?

The “package-private” or “default” , the members are accessible only within the same package only.

Accessible within the same class

Accessible within the same package

Not accessible in different package.