

ASSIGNMENT - 1

Page: _____
Date: 11/11/2023

Q1) Draw and explain the Architecture of Java

Ans Java Architecture is a collection of components i.e JVM, JRE and JDK. It integrates the process of interpretation and compilation. It defines all the processes involved in creating a Java program. Java Architecture explains each and every step of how a program is compiled and executed.

Java Architecture can be explained by using the following steps

- There is a process of compilation and interpretation in Java
- Java compiler converts the Java code and into byte code
- After that, the JVM converts the byte code into machine code
- The Machine code is then executed by machine

Java Architecture

Java Source Code files eg. A.java, B.java etc

Java Compiler

Compiled Java Source code files into Byte Code (.class) files eg. A.class, B.class etc

Java API classes used in import statements eg string, vector, etc

Class Loader

Byte Code

Execution Engine

Java Virtual Machine(JVM)

(d) Explain nine Object Oriented concepts.

Ans Different Object Oriented Concepts are as follows

① Object

An object is an instance of a class. An instance is the existence of in the computer system by acquiring a memory space

An object is the only way a class becomes usable. An object has all the characteristics of the class using which the object is created.

② Class

A class is Blueprint or stencil for creating objects. A class specifies all the member data and member functions that would be present in the objects created using class

Using a class any number of objects can be created. It's a user defined data type also called

Abstract Data Type, which acts like a basic data type when used.

③ Encapsulation

It's the process of wrapping up of Data and functions inside a single Entity (usually class).
Encapsulation helps to establish Abstraction and Data Hiding.
classes Interface and member methods implement Abstraction encapsulation at programmer level.

④ Abstraction

It's the process of Hiding the complexities of implementation and providing a simple interface for easy use.

It's implemented using Encapsulation, classes, Interface and member methods implement Abstraction at programmer level.

⑤ Data Hiding

It's the process of keeping the Data under such an Access mode that it's only accessible to permitted functions.

Using private, protected specifier, we can hide Data from access from outside a class
Classes and member methods implement Data hiding at programmer level

⑥ Inheritance

It's the process of passing attributes (data members) and characteristics (member functions) of one class to other classes. Here the class which gets the inherited properties is called the child class and from which it acquires the properties is called the base or parent class

- It forms a Hierarchical structure
- It implements reusability of Parent class data and functions
- Types - Single, Multi Level, Multiple and Hierarchical.

⑦ Polymorphism

It's the process of defining more than one kind implementation (wding) of a method, using same name with either different number of Arguments or Different Data Types of Argument.

Implementation - Method overloading and constructor overloading

- Type - Compile Time and Run Time

⑧ Dynamic Binding

It's also called Runtime polymorphism where the compiler recognizes the binding between the method call and its code during runtime. It's usually implemented using Interface Inheritance.

⑨ Message Passing

It's the process of passing data between objects of same class and also of different classes. It's implemented using member method calling.

when we pass values in the input arguments to a member method then we are passing messages to the object which is used to call the method.

when a member method returns a value then that value is the message received by us from an object that is used to call the method.

(Q3) Explain with proper example the utility of finalize() method with respect to automatic garbage collection in Java

Ans finalize() method

finalize is the method used for the purpose of garbage collection when an object goes out of scope or becomes null.

finalize method is called by the garbage collector just before destroying an object to perform cleanup activities.

finalize() method is defined in java.lang.Object

class, which means it available to all the classes for sake of overriding.

Garbage collection in Java.

One of the most important point of finalize method is that its not automatically chained like constructors. If we are overriding finalize method then its our responsibility to call finalize() method of super-class, if we forget to call it then finalize of super class will never be called. The best way to call super class finalize method is to call them in finally block using super.finalize()

Any exception thrown by finalize method is ignored by Garbage collector thread and it will not be propagated further.

There are two ways of running the finalize method, is by calling System.gc() or explicitly calling the finalize() method like a ^{member} method using the object.

Code :-

class A
{

 A() { System.out.println("Constructor A"); }

 protected void finalize() { System.out.println("Finalize"); }

{

public class Myfinalize
{

 Static void test() {

{

 A ob = new A();

 ob.finalize();

{

 • public static void main (String arg[]) {

{

 test();

 System.gc();

{

9

Output:

Constructor A

Finalize A

Finalize A