

## Assignment ①

Q1 Explain OOPs Concepts.

→ class :

class is a user defined datatype. It is template or blueprint for our object. It is a logical entity. It can also be defined as collection of objects. class doesn't store any space. Once a class is defined we can create array no. of objects belonging to that day class. The behaviour of class is similar like any other built-in data type.

Object :

Object is instance of class. It is an entity which has state and behaviour. It can be both physical and logical in nature. An object contains an address and take up some space in memory. Object can communicate with knowing detailed of each other date or code. Object represents person, place, date or any item that the program may handle. For example a dog is an object because it has states i.e. colour, name, breed etc as well as behaviour i.e. wagging the tail, barking, eating etc.

Abstraction :

Hiding internal details & showing functionally. In java we use abstract class & interface to achieve abstraction. It refers to act of representing essential features without including background details. classes are

defined as list of abstract attributes.

### Encapsulation:

Birding (or wrapping) code & data together into a single unit. Data Encapsulation is striking feature of a class. Here, the data is not accessible to the outside world & only those methods which are wrapped in class can access it. These methods provide interface between objects data & program. This insulation of data from direct access by program is called data hiding.

Ex: A capsule, it is wrapped with different medicines.

### Inheritance:

when one object acquires all the properties and behaviors of parent object. It provides code reusability. It is also used to achieve runtime polymorphism. It supports hierarchical classification. It provides idea by which we can add additional features to existing class without modifying it. This is possible by deriving new class from existing one. In Java, derived class is known as subclass. Each subclass defines only those features that are unique to it. Example: Consider a bird robin which is part of class flying bird, which is again part of class bird as shown in the below figure.

Polymorphism:  
poly means many and morph means form. so polymorphism is ability to take more than one form. when one task is performed by different ways is known as polymorphism. An operation may exhibit different behaviour in different instances. In java, we use method overloading and method overriding to achieve polymorphism. It is extensively used in implementing inheritance. Example: Consider operation of addition. For numbers, operation will generate a sum & for strings it will produce a third string by concatenation.

Q2

→ List & Explain the features of java.

Java is defined as general purpose high level computer programming language that is concurrent class-based, object-oriented & has platform independent architecture.

The following are the features of java:

~~Simple~~: Java is very easy to learn and its syntax is similar to C, C++.

~~Object Oriented~~: Everything in java is an object. It means we organize our software as a combination of different types of objects that incorporates both data and behavior.

~~Platform Independent~~: Java code can be run on multiple platforms eg: windows, linux, and Mac OS etc. Multiple platforms ie. write once and run

Anywhere (WORA).

Secure:

With Java we can develop virus-free systems. It is secured because: it has no explicit pointers, it runs inside JVM, class loader, byte code verifier, security manager.

Robust:

Because of strong memory management, automatic garbage collection, exception handling & type checking mechanism. It is resistant to failures.

Architectural Neutral:

Because there is no implementation dependent feature: size of primitive type is fixed.

Portable: Because it facilitates you to carry the java bytecode to any platform. It doesn't require any type of implementation.

High Performance:

Java is faster than other traditional interpreted programming languages because java bytecode is "close" to native code.

Interpreted:

Java interpreter is high quality, high performance & less space occupying. Making it one of the fastest in current scenario.

Distributed :

Because it facilitates users to create distributed applications in java. RMI and EJB are used for creating distributed applications.

Multithreaded :

A thread is like a separate program. The advantage is that it doesn't occupy memory for each thread. It shares a common memory area.

Dynamic :

Java is a dynamic language - It supports dynamic loading of classes. It means classes are loaded on demand. It also supports functions from C & C++.

Q3 Explain method Overloading with programming example.

- It is process of creating methods with same name, but different parameters list & different definitions.
- It is used when objects are required to perform similar task with different input parameter.
- When we call a method, Java matches up the method name first & then type of parameters to decide which method to execute.
- Following example shows concept of method overloading.

class Overload

{  
void test()  
{

; System.out.println("No parameters");

void test(int a)  
{

; System.out.println("a"+a);

void test(int l, int m)  
{

; System.out.println("l"+l+" and "+m);  
+m);

double test(double x)  
{

; System.out.println("x:"+x);

return(x\*x);

class Overload Demo

public static void main (String args [])

Overload obj = new Overload();

obj.test();

obj.test(3);

obj.test(11, 12);

double result = obj.test(6.75);

System.out.println("Result = "+result)

Q4 what is Constructor? Explain with programming example.

→ Constructors

- Constructors enables an object to initialize itself when it is created.
- It has same name as that of class itself.
- It does not have any return type not even void.
- They return the instance of class itself.

Types of constructor

Default Constructor

Parameterized Constructor

- Default Constructor → If a constructor does not accept any parameters it is known as a default constructor.
- Parameterized Constructor → If takes argument to initialize the object attributes.

Syntax:

public class className

{     public className() // Constructor  
        {     // Constructor body     }

Example:

Class Rectangle

int length, width;

Rectangle (int x, int y) // Constructor method  
: parameterized

{  
length = x;  
width = y;

int rectArea()

{  
return (length \* width);  
}

Class RectangleArea.

public static void main (String args [])

{  
Rectangle rect = new Rectangle (5, 10);  
// calling constructor.

int area = rect.rectArea();

System.out.println ("Area is " + area)

Q5

Explain the java program structure.

Java Program Structure

Documentation section

Package statement

Import statement

Interface statement

Class Definition

Main Method class

//Main Method definition

- Documentation Section - It includes the comments to tell the programs purpose. It involves improves the readability of the program. It basically comprises a set of comment lines giving the name of the program the author and the other details.
- Package Statement - The first statement allowed in java file is package statement. This statement declares a package name & informs the classes that the classes defined here belongs to this package.

Ex - package Student;

- Import Statement - The next thing after a package statement may be a number of import statement. This is similar to #include

Statement in C.

Example

Import Student. test;

This statement instructs the interpreter to load test class contained in the package Student.

- ④ Interface Statement → An interface is like class but includes a group of method declaration. This is also an optional section and used when we wish to implement the multiple inheritance feature in the program.
- Class Definition - A Java program may contain multiple class definition. Classes are the primary and essential element of Java programs. These classes are used to map the objects of real world problems. The number of classes used depends on the complexity of the problem.
- Main Method class - Since every Java stand-alone program requires a main method as its starting point. This class is the essential part of the Java program. A simple Java program may contain only this part. This main method creates object of various classes & establishes communication between them. On reaching the end of main, the program terminates and control passes to the operating system.

## Example Program . . .

class Sample

public static void main (String args [])

{  
    System.out.println ("welcome to java")  
}

Print