ASSIGNMENT-1 SET-3

JAVA PROGRAMMING

1. What is data abstraction? Differentiate data and procedural abstractions. write inheritance hierarchy for the superclass Quadrilateral, parallelogram, square and Rectangle, calculate area of the square, Rectangle and parallelogram.

Data Abstraction:

Data Abstraction is the process of hiding certain details and showing only essential information to the user, Abstraction can be achieved with either abstract classes or interfaces.

eg: A car is viewed as a car rather than its individual components.

Data Abstraction may also be defined as the process of identifying only the required of identifying only the required characteristics of an object ignoring the irrelevant details. The properties and behaviours of an object differentiate it from other objects of similar type and also help in classifying/grouping the objects.

Procedural Abstractions are normally characterized in a programming language as "function/sub-function" or "procidure abstraction. Data abstraction, instead of just focusing on operations, we focus on data first and then operations that manipulate the data.

In procedural Abstraction, methods are used to capture the procedural patterns, abstracting over behaviour.

Principles of OOP:

The following are the 3 types of OOPs (principles).

1. Encapsulation
2. Inheritance
3. Polymorphism.

1. Encapsulation: Wrapping up of the data and methods into a single unit is known as Encapsulation. Encapsulation is the mechanism that binds together code and the data it manipulates, and keeps both safe from outside interference and misuse.

In Java the basis of encapsulation is the class. A class defines the structure and behaviour that will be shared by a set of objects. Thus, a class is a logical construct, an object has physical reality.

2. Inheritance: Inheritance is the process by which objects of one class acquire the properties of objects of another class. A class inherits state and behaviour from its super-class. Inheritance provides a powerful and natural mechanism for organizing and structing software programs.

Reusability is main aduantage in inheritance by which we can add additional features to an existing class. without modifying it.

3. Polymorphism: polymorphism means the ability to take more than one form. For example, an operation may exhibit different behaviour in different instances. The behaviour depends upon the types of data used in the operation. For example, consider the operation of addition. For two numbers, the operation will generate a sum. If the operands are strigs, then the operation would produce a third string by concatenation.

Program:

```
Public class QuadrilateralTest {
Public static void main(String[] args){
 // All co-ordinates are assumed to form the proper shapes
 // A Quadrilateral is a four-sided polygon
 Quadrilateral quadrilateral =
 new Quadrilateral(1.1, 1.2, 6.6, 2.8, 6.2, 9.9, 2.2, 7.4);
 // A parallelogram is a quadrilateral with opposite sides
                                                  Parallel
 Parallelogram parallelogram =
 new Parallelogram(5.0, 5.0, 11.0, 5.0, 12.0, 20.0, 6.0, 20.0);
 // A rectangle is an equiangular parallelogram
 Rectangle rectangle=
 new Rectangle(17.0, 14.0, 30.0, 14.0, 30.0, 28.0, 17.0, 28.0);
 // A square is an equiangular and equilateral parallelo-
                                                  -gram
 Square square =
 new Square(4.0, 0.0, 8.0, 0.0, 8.0, 4.0, 4.0, 4.0);
```

```
System.out.printf(
"%s%s%s%s\n", quadrilateral, parallelogram,
rectangle, square);
}
}
```

Output:

Coordinates of Quadrilateral are:

(1.1, 1.2), (6.6, 2.8), (6.2, 9.9), (2.2, 7.4)

Coordinates of Parallelogram are:

(5.0, 5.0), (11.0, 5.0), (12.0, 20.0), (6.0, 20.0)

Width is : 6.0

Height is : 15.0

Area is : 90.0

Coordinates of Rectangle are:

(17.0, 14.0), (30.0, 14.0), (30.0, 28.0), (17.0, 28.0)

Width is : 13.0

Height is : 14.0

Area is : 182.0

Coordinates of Square are:

(4.0, 0.0), (8.0, 0.0), (8.0, 4.0), (4.0, 4.0)

Side is : 4.0

Area is : 16.0

2. what is the importance of constructor? write a java program on constructor overloading. Describe the usage of static members and nesting members with suitable example in java programs.

Constructor: It is a block of code that initializes the newly created object. A constructor resembles an instance method in java but it's not a method as it doesn't have a return type.

Importance of constructor:

The purpose of constructor is to initialize the object of a class while the purpose of a method is to perform a task by executing java code. Constructors cannot be abstract, final, static and synchronised while methods can be constructors do not have return types while methods do. In brief, constructor and method are different (more on this at the end of this guide). People often refers constructor as special type of method in Java.

Java Program on Constructor Overloading:

Constructor Overloading is a concept of having more than one constructor with different parameters list, in such a way so that each constructor performs a different task.

```
public class Demo{
  Demo(){
  ..
  }
  Demo(String s){
  ....
  }
  Demo(Inti){
  ....
  }
....
}
```

Three overloaded Constructors - They must have different Parameters list.

Example :

```
//Student.java

class Student
{
  int Roll;
  String Name;
  double Marks;

  Student(int R, string N, double M)   // constructor1
  {
    Roll=R;
    Name=N;
    Marks=M;
  }

  Student(String N, double M, int R)   // constructor2
  {
    Roll=R;
    Name=N;
    Marks=M;
  }

  void Display()
  {
    System.Out.print("\n\t"+Roll+"\t"+Name+"\t"+
                                        Marks);

  }
}
class ConstructorOverloadingDemo
{
```

```
Public static void main(String[] args)
{
    Student S1 = new Student (1, "kumar", 78.53); // statement 2
    Student S2 = new Student ("Sumit", 89.42, 2); // statement 1

    System.out.print("\n\t Roll\t Name\t Marks\n");
    S1.Display();
    S2.Display();
}
}
```

Output:

Roll	Name	Marks
1	kumar	78.53
2	Sumit	89.42

In the above example, we have two constructors.
Statement 2 will invoke constructor 1 because the signature of Constructor 1 is similar to the statement 2.
Statement 1 will invoke constructor 2 because the signature of constructor 2 is similar to the statement 1.

Static members and nesting members:

A static class i.e., created inside a class is called static nested class in Java. It cannot access non-static data members and methods. It can be accessed by outer class name.

→ It can access static data members of outer class including private.

→ static nested class cannot access non-static (instance) data member or method.

Static nested classes usage:

when you need to write many and smaller classes.

Instead of creating many small program files, you make one program file with all the tiny static classes.

Java static nested class Example with static method:

If you have the static member inside static nested class, you don't need to create instance of static nested class.

```
class TestOuter↵{
  static int data=30;
  Static class inner {
  Static void msg()
{ system.out.println("data is +" data); }.
}
Public static void main (String args[])
TestOuter↵.Inner.msg(); // no need to create the
instance of static nested class.
}
}
```

Output:

Compile by: Java TestOuter↵.java

Run by: Java TestOuter↵

data is 30

3. Define a class named Book-fair with the following description:

Instance variables / Data members:

String Bname – stores the name of the book.

double price – stores the price of the book.

Member Methods:

(i) Book-fair( ) – Default constructor to initialize data members.

(ii) Void Input( ) – To input and store the name and the price of the book.

(iii) Void calculate( ) – To calculate the price after discount.

Discount is calculated based on the following criteria.

Price	Discount
Less than or equal to Rs.1000	2% of price
More than Rs.1000 and less than or equal to Rs.2000	10% of price
More than Rs.3000	15% of price

(iv) Void display( ) – To display the name and price of the book after discount.

```
import java.io.*;
class book-fair
{
 //instance variables
  String bname;
  double p;
  double np;
  book-fair( )
  {
   bname = "";
    p = 0.00f;
   np = 0.00f;
  }
```

```
void input( ) throws IOException
{
 Buffered Reader br=new
 Buffered Reader(new
 input Stream Reader(System.in));
 System.out.println("enter the book name");
 bname = br.readline();
 System.Out.println("enter the price");
 P=Double.parse Double(br.readline());
}
void calculate()
{
 if(P<=1000)
  {
   np=p-(P*0.2);
  }
 else if(p>1000 && P<=3000)
  {
    np=p-(P*0.1);
  }
 else
  {
    np=p-(P*0.15);
  }
}
void display()
{
 System.Out.println("book name\t" + "price\t"+"netpiece
 System.Out.println(bname+"\t"+p+"\t"+np);
}
```

```
    Public Static Void main(String[] args)
    throws IOException
    {
      bookfair obj= new bookfair();
      Obj.input();
      obj.calculate();
      obj.display();
    }
}
```

4. write a main method to create an object of the class and call the above member methods.

· Special words are those words which starts and ends with the same letter.

Examples:

EXISTENCE
COMIC
WINDOW

Palindromes are special words which read the same from left to right and vice-versa.

Examples:

MALAYALAM
MADAM
LEVEL
ROTATOR
CIVIC

All palindromes are special words but all special words are not palindromes.

write a program to accept a word check and print whether the word is a palindrome or only special word.

19BQ1A05M4


```
public class Palindrome:
{
 Public static void main (string args[])
 {
  String a,b = "";
  scanner s = new Scanner (system.in);
  scanner (systemin);
  System.out print("Enter the string you want
                          to check:");
  a = s.nextline( );
  int n =a.length( );
 for(int i=n-1; i>=0; i--)
 {
  b=b+a.charAt(i);
 }
  if (a.equalsIgnorecase(b))
  {
   system.out.println ("It is only special word");
  }
  system.out
  {
   system.Out.println("It is only not special word");
  }
 }
}
```

Output: Enter the string you want to check : NeverOddorEven
The string is palindrome.