Regd. No: 19BQ1A05K4
SEC: CSE-D

SET-3 JAVA ASSIGNMENT-I

1) What is data abstraction? Differentiate data and procedural abstractions. Write inheritance hierarchy for the super class Quadrilateral, Parallelogram, Square and Rectangle. Calculate area of square, rectangle and parallelogram.

A) Abstraction means displaying only essential information and hiding the details. Data abstraction refers to providing only essential information about the data to the outside world, hiding the background details or implementation. Consider a real life example of a man driving a car. The man only knows that pressing the accelerators will increase the speed of car or applying brakes will stop the car but he does not know about how on pressing the accelerator the speed is actually increasing, he does not know about the inner mechanism of the car or the implementation of accelerator, brakes etc in the car.

Procedural Abstraction:

Procedural abstraction provides mechanisms for abstracting well defined procedures or operations as entities. The implementation of the procedure requires a number of steps to be performed. A simple example is a debit operation which performs various steps to debit certain amount from the bank account. Hence at the banking level, credit and debit become well defined procedural abstractions. These are extensively used by requirements analysts, as well as designers and programmers.

Procedural abstractions are normally characterized in a programming language as "function/sub-function" or "procedure" abstraction.

Data Abstraction:

This principle is at the core of Object Orientation. In this form of abstraction, instead of just focusing on operations, we focus on data first and then the operations that manipulate the data. A simple example is queue data and the associated operations add() and delete(). Both add() and delete() operations manipulates queue data. In a simple procedural abstraction, there would be only add and delete operations separately but their association with the queue data will not be captured. The advantage of data abstraction over procedural abstraction is that the data and the associated operations get specified together and hence it is easy to modify the code when data changes.

Inheritance hierarchy for the super class Quadrilateral, Parallelogram, Square and Rectangle:

```
import java.util.Scanner;
class QuadrilateralTest {
    double x;
    double y;
    Scanner sc=new Scanner(System.in);
}
class Parallelogram extends QuadrilateralTest{
   public void input(){
       System.out.println("Enter the side1 value of Parallelogram");
       x= sc.nextDouble();
       System.out.println("Enter the side2 value of Parallelogram");
       y=sc.nextDouble();
   }
   public double area(){
       return x*y;
}
}
```

```
class Rectangle extends QuadrilateralTest{
    public void input(){
        System.out.println("Enter the side1 value of Rectangle");
        x= sc.nextDouble();
        System.out.println("Enter the side2 value of Rectangle");
        y=sc.nextDouble();
    }
    public double area(){
        return x*y;
    }
}
class Square extends QuadrilateralTest{
    public void input(){
        System.out.println("Enter side value of Square");
        x=sc.nextDouble();
    }
    public double area(){
        return x*x;
    }
}
public class Quadrilateral{
    public static void main(String args[]){
        Parallelogram ob1 = new Parallelogram();
        Rectangle ob2= new Rectangle();
        Square ob3=new Square();
        ob1.input();
```

```
        System.out.println("\n Area of the Parallelogram " + ob1.area()+"\n");
        ob2.input();
        System.out.println("\n Area of the Rectangle "+ob2.area()+"\n");
        ob3.input();
        System.out.println("\n Area of the Square "+ob3.area()+"\n");
    }
}
```

Output:

```
Enter the side1 value of Parallelogram
5
Enter the side2 value of Parallelogram
4
Area of the Parallelogram 20.0

Enter the side1 value of Rectangle
6
Enter the side2 value of Rectangle
3
Area of the Rectangle 18.0

Enter side value of Square
5
Area of the Square 25.0
```

2) What is the importance of constructor? Write a java program to perform constructor overloading. Describe the usage of static members and nesting members with suitable example programs in java.

A) In simple word, Constructor is a method like a block of code which is called by Java runtime during object creation using new() operator. Constructor are special in the sense that they have the same name as the Class they are part of. They are also special in a sense that they are called by JVM automatically when you create an object.

Importance of Constructor:

One reason is to initialize your object with default or initial state since default values for primitives may not be what we are looking for. One more reason we create constructor is to inform the world about dependencies, a class needs to do its job. Anyone by looking at our constructors should be able to figure out, what he needs in order to use this class.

Constructor can be overloaded:

This means we can have more than one constructor in our class (all with the same name) until they have different method signature, which comprises type of argument and order type of argument. Here is an example of constructor overloading. Here we have three constructors but all with a different set of parameters.

Example of Constructor Overloading:

```
//Java program to overload constructors
class Student5{
    int id;
    String name;
    int age;
    //creating two arg constructor
    Student5(int i, String n){
    id = i;
    name = n;
    }
    //creating three arg constructor
    Student5(int i, String n, int a){
    id = i;
    name = n;
    age = a;
    }
    void display()
    {System.out.println(id+" "+name+" "+age);}
    public static void main(String args[]){
    Student5 s1 = new Student5(111, "Karan");
    Student5 s2 = new Student5(222, "Aryan",25);
    s1.display();
    s2.display();
    }
}
```

Output:

```
111 Karan 0
222 Aryan 25
```

Static Members in Java:

In Java, static members are those which belongs to the class and we can access these members without instantiating the class. The static keyword can be used with methods, fields, classes (inner/nested), blocks.

Use of static members in Java, when a member is declared static, it can be accessed before any objects of its class are created, and without reference to any object.

Example of static method:

```
class Test
{
    //static method
    static void m1()
    {
      System.out.println("from m1");
    }
    public static void main(String[] args):
    {
        m1();
    }
}
```

Output:

```
from m1
```

Nesting members in Java:

They enable you to logically group classes that are only used in one place, thus this increases the use of encapsulation, and creates more readable and maintainable code. As a member of its enclosing class, a nested class can be declared private, public, protected, or package private (default).

Example of Static Nested class:

```
Class OuterClass
{
    static int outer_x = 10;
    int outer_y = 20;
    private static int outer_private = 30;
    static class StaticNestedClass
    {
        void display()
        {
            System.out.println("outer_x = " + outer_x);
            System.out.println("outer_private = " + outer_private);
        }
    }
}
public class StaticNestedClassDemo
{
    public static void main(String[] args)
    {
        OuterClass.StaticNestedClass nestedObject = new OuterClass.StaticNestedClass();
        nestedObject.display();
    }
}
```

Output:

```
outer_x = 10
outer_private = 30
```

3) Define a class named BookFair 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) BookFair() - 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 3000	10% of price
More than Rs 3000	15% of price

(iv) void display() - To display the name and price of the book after discount.

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

A) Program

```
import java.util.Scanner;
class BookFair{
    String Bname;
    double price;
    Scanner sc = new Scanner(System.in);
    public BookFair(){}
```

```
public BookFair(String Bname, double price){
    this.Bname = Bname;
    this.price = price;
}
public void Input(){
    System.out.println("Enter Book Name: ");
    Bname = sc.nextLine();
    System.out.println("Enter Book Price: ");
    price = sc.nextDouble();
}
public void calculate(){
    if (price<=1000.00){
        price = price - (price * 0.02);
    }
    else if(price>1000.00 && price<=3000.00){
        price = price - (price * 0.1);
    }
    else{
        price = price - (price * 0.15);
    }
}
public void display(){
    System.out.println("Name of the book = " +Bname);
    System.out.println("Price of the book after discount = " +price);
}
}
```

```
public class BKfair{
    public static void main(String args[]){
        Book fair book1;
        book1 = new Bookfair();
        book1.Input();
        book1.calculate();
        book1.display();
}
}
```

Output 1:

```
Enter Book Name:
royal
Enter Book Price:
2322
Name of the book = royal
Price of the book after discount = 2089.8
```

Output 2:

```
Enter Book Name:
real men
Enter Book Price:
678
Name of the book = real men
Price of the book after discount = 664.44
```

Output 3:

```
Enter Book Name:
jackpot
Enter Book Price:
4554
Name of the book = jackpot
Price of the book after discount = 3870.9
```

4) Special words are those words which starts and ends with the same letters.

Examples:

EXISTENCE

COMIC

WINDOW

Palindrome words are those 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.

A) Program

```
import java.util.*;
public class Words
{
    public static void main(String[] args) {
        String original, reverse = "";
        Scanner in = new Scanner(System.in);
        System.out.println("Enter a string to reverse: ");
        original = in.nextLine();
        int length = original.length();
```

```
        for (int i=length-1; i>=0; i--)
            reverse = reverse + original.charAt(i);
        System.out.println("Reverse of the string: " + reverse);
        if (original.equals(reverse)&&(original.substring(0,1).equals(original.substring
          (length-1))))
            System.out.println("Palindrome");
        else if (original.substring(0,1).equals (original.substring(length-1)))
            System.out.println("Special word");
        else
            System.out.println("None");
    }
}
```

Output 1:

```
Enter a string to reverse:
MADAM
Reverse of the string: MADAM
Palindrome
```

Output 2:

```
Enter a string to reverse:
COMIC
Reverse of the string: CIMOC
Special word
```

Output 3:

```
Enter a string to reverse:
ROTATOR
Reverse of the string: ROTATOR
Palindrome
```