19BQ1A05G7 Set-3

1) Data abstraction is a process of hiding certain data and showing only esential information to the user. In this form of showing only esential data, we focus on data first and then the operations that manipulate the data. The product of data abstraction in type (ADT). In OOPS languges like Java, ADT's are implement as class

Difference between data and procedural abstractions
Procedural abstraction are normally characteried in a programming language as "function/sub-function" or abstraction. It is tied to the idea that each particular method performs a well. specified

Eg:-
```
String str = "Hi How";
string str1 = str.substring(0, 3);
```

Data Abstraction:-
In this form of abstraction, instead of just focusing on operation, we focus on data

In data Abstraction it means while designing defining the class itself you need to identify only those atributes of a class which are relevant

Program :-

```
import java.util.Scanner;
abstract class Quadrilateral{
        Public abstrac double area(doublelength, doble breath);

}
class Rectangle extends Quadrilateral{
        Public double (double length, double breath){
                return length * breath;
        }
}
Class Square extends Quadrilateral{
        Public double area(double length, double breath){

                return length*breath;

}
class Parllalogram extends Quadrilateral{
        Public double area(double length, double breath){
                return length * breath;}

}
Public class Cal{
        Public static void main(string[] args){
            Square -sq = new square();
            system.out.println(sq.area(2,2));
            Rectangle r1 = new Rectangl()
            system.out.Println(r1.area(2,3));}}
```

2) Constructor is verry important to restrict the data. It ~~does~~ help in initializing the private variables. The another main use by the Constructor is to initialize default values. The memory is also allocated by the default Constructors. By default the compiler ~~uses~~ uses default Constructor until a user-defined constructor is added. The Constructor name is as same as class name

Eg :-

```
class Const{
    String str;
    Private String str1;
    Const( ){ }
    Const (String str){
        this.str= str;
    }
    Const (String str, String str1){
        this.str1=str1;
        this.str= str;
    }
```

Static members:-

Mainly static is used to save memory. It is not changed for every object. It is Constant for the whole class. It does not change when ever you change the object by this property there will be a efficient space saving. when it Comes to method we can access one static method with another static method only. If the value of static member is changed using an object then the value is changed for the other objects also

Eg:-
```
class stat{
        static int defau=0;
        ~~stat(){~~
        Public void count(){
            defau++;
        }
}
Public class Main{
    Public static void main(String[] args){
        Stat obj1 = new stat();
        system.out.println(~~obj1.count~~ obj1.defau);
        staf obj2 = new stat();
        system.out.println(obj2.defau);
```

Nesting members :-

A nested class is a member of its enclosing class. The advantage of nested members is they can acess to the other members of the enclossing class if they are private also there condition is they shouldn't be static.

Eg :-

```
Public class Main {
        int a = 0
        class one{
            int a=0;
          Public void one_method(int x){
                system.out.println(x);
                System.out.println(this.x);
           }
        }
        Public static void main(String[] args){
          Main obj = new Main();
            obj.one obj1 = new obj.one();
              obj1.one_method(5);

        }
}
```

3

```
import java.utill.Scanner;
public class discount{
        Public static void main(String[]args){
            BookFair obj = new BookFair();
            obj.input();
            obj.calaculate();
            obj.display();
        }
}
class BookFair{
        string Bname;
        double price;
        BookFair(){ }.
        Public Void input(){
        Scanner inp= new scanner(system.in);
          Bname = inp.nextLine();
          Price =  inp.nextInt();
        }
        Public Void calaculate(){
           if(Price>1000 && Price<300){double s=100-10; Pire=
                                                        (sxPrice/100)}
           if(Price>3000){double S=100-15; Price =(S*price)/100);}
           if(Price<=1000){double S= 100-2; Price=(sxprice)/100);}
        }
        Public void display(){
           System.out.println(" The final price of product":price);

        }
}
```

4)
```
Public class special{
        public static void main(String[] args){



        }
       Public static boolean specialchar(string ·str){
              int len = str.length-1;
              if(str.chatAt(0) == str.charAt(len))
                     return true;
              else
                 return false;
       }
       Public static boolean palindrome(string str{
              String ne=" ";
              for(int i=str.length()-1; i>=0; --i){
                  ne= ne+str.charAt(i);
              }
              if(str.equal(ne))
                  retur true;
              else
                  return false;
```