1.Stack Operations using Interface: Create an interface Stack with a variable size and abstract methods push(), pop(), display(), overflow(), and underflow(). Implement a subclass IntegerStack that implements the Stack interface. Create a test class to check the working of all methods in the IntegerStack class.

Code

```
import java.util.*;
interface Stack{
    public static int size=5;
    public void push();
    public void pop();
    public void display();
}
class IntegerStack implements Stack{
    int top=-1;
    int [] arr=new int[size];
    int num;
    Scanner sc=new Scanner(System.in);
    public void push(){

        if(top==size-1){
            System.out.println("Stack is full");
        }else {
            top++;
            System.out.println("Enter the element to push");
            num=sc.nextInt();
            arr[top]=num;
    }
}
public void pop(){
    if(top==-1){
        System.out.println("Satck is empty");
    }else{
        System.out.println("Element popped is" +
arr[top]);
        top--;   }
}
public void display(){
    if(top==-1){
        System.out.println("Stack is empty");
    }else{
        for(int i=top;i>=0;i--){
            System.out.println(arr[i]);
        }}}}
class StackOp{
    public static void main(String[]args){
        IntegerStack s=new IntegerStack();
        Scanner sc=new Scanner(System.in);
        int ch;
        while (true) {
            System.out.println("1.push 2.pop 3.display 4.exit");
            System.out.println("Enter your Choice");
            ch=sc.nextInt();
            switch(ch){
                case 1:
                { s.push();
                 break;}
                case 2:
                {s.pop();
                break;}
                case 3:
                s.display();
                break;
                case 4:
                System.out.println("exit");
                }
 }}}
```

OUTPUT:-

1.push 2.pop 3.display 4.exit

Enter your Choice

1

Enter the element to push

2

1.push 2.pop 3.display 4.exit

Enter your Choice

1

Enter the element to push

2

1.push 2.pop 3.display 4.exit

Enter your Choice

1

Enter the element to push

5

1.push 2.pop 3.display 4.exit

Enter your Choice

1

Enter the element to push

8

1.push 2.pop 3.display 4.exit

Enter your Choice

3

Elements in stack:

8

5

2

2

1.push 2.pop 3.display 4.exit

Enter your Choice

2

Element popped is8

1.push 2.pop 3.display 4.exit

Enter your Choice

4

exit

2.Shape Interface with Rectangle and Triangle: Implement the following:

a.Create an interface Shape with an abstract method area().

b.Create two classes, Rectangle and Triangle, that implement the Shape interface.

Calculate and display the area of both Rectangle and Triangle

code

```
interface Shape{
   public static final double PI=3.14;
   public double area(double d1,double d2);
}
```

```
class Rectangle implements Shape{
    public double  area(double x,double y){
        return x*y;
    }
}
class Circle implements Shape{
    public double area(double x,double y){
        return PI*x*y;
    }
}
class Triangle implements Shape{
    public double area(double x,double y){
        return x*y;
    }
}
public class Area4 {
   public static void main(String[] args){
    Shape s;
    Rectangle r=new Rectangle();
    s=r;
    System.out.println("Area of Rectangle is" + s.area(6,3));
    Circle c=new Circle();
    s=c;
    System.out.println("Area of Circle is" + s.area(3,3));
    Triangle t=new Triangle();
    s=t;
    System.out.println("Area of Triangle is" + s.area(9,5));
 }
}
```

OUTPUT

Area of Rectangle is18.0

Area of Circle is28.259999999999998

Area of Triangle is45.0

3.Student Exam Results Using Inheritance and Interface in: Implement the following hierarchy:

a.Create a class Student with a variable rollNo and methods getRollNo() and setRollNo().

b.Create a class Test that inherits Student and has variables sub1 and sub2 with methods getMarks() and setMarks().

c.Create an interface Sports with a variable sMarks and a method set().

d.Create a class Result that inherits Test and implements the Sports interface. It should display the marks.

e.Demonstrate the functionality of these classes in a test application.

code

```
import java.util.*;
interface Sport{
   public static int smarks=5;
}
class student{
   int roll_no;
   void get_roll(){
      System.out.println("Enter Roll NO:");
      Scanner sc=new Scanner (System.in);
      roll_no=sc.nextInt();

   }
   void set_roll(){
      System.out.println("Roll No is:" + roll_no);
   }
}
class Test extends student{
   double sub1,sub2;
   void get_marks(){
      Scanner sc=new Scanner (System.in);
      System.out.println("Enter marks out of 50:");
System.out.println(" Subject 1:");
      sub1=sc.nextInt();
      System.out.println(" subject 2:");
      sub2=sc.nextInt();
   }
   void set_marks(){
      Scanner sc=new Scanner (System.in);
      System.out.println(" Marks of Sub 1:" +
sub1);
 System.out.println(" Marks of sub 2:" + sub2);
}
}
class Result extends Test implements Sport{
   double total;
void display(){
 System.out.println("Is there any sport
achievement?yes/no");
   Scanner sc=new Scanner(System.in);
   String sport=sc.nextLine();
if(sport.equals("yes")){
      System.out.println("Sport marks:" +
smarks);
      total=sub1+sub2+smarks;
   }
```

```
else{
        total=sub1+sub2;}
        System.out.println("Total Marks:" + total);
    }
}
public class StudentResult {
    public static void main(String[]args){
        Result r=new Result();
        r.get_roll();
        r.get_marks();
        System.out.println("----------");
        r.set_roll();
        r.set_marks();
       r.display(); }
    }
```

OUTPUT

```
Enter Roll NO:
57
Enter marks out of 50:
 Subject 1:
45
 subject 2:
43
----------
Roll No is:57
 Marks of Sub 1:45.0
 Marks of sub 2:43.0
Is there any sport achievement?yes/no
yes
Sport marks:5
Total Marks:93.0
```