

String & Abstract

16/1/24

1) Write a java program to create a generic class stack which holds 5 integer & 5 double values.

```
Class GenericStack<T>{
```

```
private Object[] stackArray;
```

```
private int top=-1;
```

```
private static final int N=5;
```

```
public GenericStack()
```

```
{
```

```
stackArray = new Object[N];
```

```
}
```

```
public void push (T value)
```

```
{
```

```
if (top < N-1)
```

```
stackArray [top++] = value
```

```
else
```

```
System.out.println("stack is full");
```

```
}
```

```
public T pop()
```

```
{
```

```
if (top >= 0)
```

```
return (T) stackArray [top--];
```

```
else
```

```
{
```

```
System.out.println("stack is empty");
```

```
return null;
```

```
}
```

```
}
```

```
public boolean isEmpty()
```

```
{
```

```
return top == -1;
```

```
public boolean isFull()
```

```
{ return top == N-1;
```

```
}
```

```
class smain
```

```
{
```

```
public static void main (String args[])
```

```
{
```

```
GenericStack<Integer> integerStack = new GenericStack<>();
```

```
GenericStack<Double> doubleStack = new GenericStack<>();
```

```
for (int i=1; i<=5; i++){
```

```
integerStack.push(i);
```

```
}
```

```
for (double i=1.0; i<=5.0; i++){
```

```
doubleStack.push(i);
```

```
}
```

```
System.out.println("popped integers from the stack:");
```

```
while (!integerStack.isEmpty())
```

```
System.out.println(integerStack.pop());
```

```
}
```

```
System.out.println("popped double from the stack:");
```

```
while (!doubleStack.isEmpty())
```

```
System.out.println(doubleStack.pop());
```

```
}
```

```
}
```

Output:

popped integers from the stack:

5

4

3

2

1

popped doubles from the stack:

5.0

4.0

3.0

2.0

1.0

Shreyas.k

1BM22CS271

16/1/24

2) Strings - Demonstrate string length, string literal, string concat.

```
public class string1 {
    public static void main (String args[])
    {
        System.out.println ("Demo string length:");
        String a = "Hello";
        System.out.println ("a.length()");

        System.out.println ("String concat");
        String age = "9";
        String msg = "He is " + age + " years old.";
        System.out.println (msg);

        System.out.println ("Demo literals");
        System.out.println ("abc.length()");
    }
}
```

output:

Demo string length:

5

str concat:

He is 9 years old

Demo literals

3

Shreyas.k

IBM22CS271

Abstract Class.

3) create subclasses Circle & Triangle that extends the shape class

```
import java.lang.Math;
```

```
abstract class shape {
```

```
    double a;
```

```
    double b;
```

```
    double c;
```

```
    abstract void calculateArea();
```

```
    abstract void calculatePeri();
```

```
}
```

```
class triangle extends shape {
```

```
    Triangle (double x, double y, double z)
```

```
{
```

```
    a = x;
```

```
    b = y;
```

```
    c = z;
```

```
}
```

```
void calculateArea()
```

```
{
```

```
    double s = (a+b+c) / 2;
```

```
    System.out.println("Area = " + (Math.sqrt(s * (s-a) * (s-b) * (s-c))));
```

```
}
```

```
void calculatePeri()
```

```
{
```

```
    Sys.out.println("Peri = " + (a+b+c));
```

```
}
```

```
}
```



```

class Circle extends Shape {
    circle (double r)
    {
        a=r;
    }
    void calculateArea ()
    {
        Sys.out.println("Area="+Math.PI*a*a);
    }
    void calculatePeri ()
    {
        sout. ("Peri="+(2*Math.PI*a));
    }
}

```

```

class ShapeM {
    public static void main (String [], args)
    {
        Triangle t = new Triangle(2.0, 3.0, 5.0);
        circle c = new Circle(5.0);
        t.calculateArea();
        t.calculatePeri();
        c.calculateArea();
        c.calculatePeri();
    }
}

```

Op:- Area = 4.145
 Peri = 11.0
 Area = 78.53981
 Peri = 31.4159.

Shreyas.K
 IBM22CS271