| | | |
|---|---|---|
| **Name of Student:** Pushkar Prasad Sane | | |
| **Roll Number:** 45 | | **Lab Assignment Number:** |
| **Title of Lab Assignment:** Assignment based on Generics | | |
| **DOP:** 31/08/2023 | | **DOS:** |
| **CO Mapped:** | **PO Mapped:** | **Signature:** |

## PRACTICAL 3

**Aim:**

**A)** Implement bounded types (extend super class) with generics.
Create a class shape with method Area() create circle and Square which extends Class Shape.
Create a generic class Bounded Shape that extends shape and implement the generics and use area function accordingly.

**Code:**

```
package test;

abstract class Shape{
	double d;
	abstract double area();
}
class Circle extends Shape{
	Circle(double d1){
		d = d1;
	}
	double area() {
		return Math.PI * d * d;
	}
}
class Square extends Shape{
	Square(double d1){
		d = d1;
	}
	double area() {
		return d * d;
	}
}
class BoundedShape<T extends Shape>{
	T ob;
	BoundedShape(T ob1){
		ob = ob1;
	}
	void area(String shape) {
		System.out.println("Area of " + shape + " = " + ob.area());
	}
}
public class test {
	public static void main(String[] args) {
		Circle c = new Circle(5);
```

```
                BoundedShape<Circle> boundedcircle = new
BoundedShape<Circle>(c);
                boundedcircle.area("Circle");

                Square s = new Square(5);
                BoundedShape<Square> boundedsquare = new
BoundedShape<Square>(s);
                boundedcircle.area("Square");
        }
}
```

**Output:**

```
Area of Circle = 78.53981633974483
Area of Square = 78.53981633974483
```

**B)** Implement bounded types (implements an interface) with generics. Create an Interface shape with method Area() create Circle and Square which implements Class Shape. Create a generic class Bounded Shape that extends shape and implement the generics and use area function accordingly.

**Code:**

```
interface Shape{
   double area();
}
class Circle implements Shape{
   double radius;
   Circle(double d){
      radius=d;
   }
   public double area(){
      return Math.PI * radius * radius;
   }
}
class Square implements Shape{
   double side;
   Square(double d){
      side=d;
   }
   public double area(){
      return side * side;
   }
}
class BoundedShape<T extends Shape>{
   T ob;
   BoundedShape(T ob1){
      ob=ob1;
   }
   void area(String shape){
      System.out.println("area of "+ shape +" is = "+ ob.area());
   }
}
public class Main{
        public static void main(String[] args) {
           Circle c=new Circle(2);
                BoundedShape<Circle> boundedCircle=new BoundedShape<Circle>(c);
                boundedCircle.area("circle");

                Square s=new Square(2);
```

```
                BoundedShape<Square> boundedSquare=new
BoundedShape<Square>(s);
                boundedSquare.area("square");
        }
}
```

**Output:**

```
PS F:\Pushkar\MCA\Sem 1\Java>  & 'C:\Program Files\Java\jdk-20\
cbad9d3df6\redhat.java\jdt_ws\Java_75c21111\bin' 'Practical3B'
area of circle is = 12.566370614359172
area of square is = 4.0
PS F:\Pushkar\MCA\Sem 1\Java>
```