

Ex. 4 ABSTRACTION AND POLYMORPHISM

Date: 09-08-2024

PROGRAM 1

AIM:

To create a Java program that demonstrates the concept of abstract classes and inheritance by defining an abstract class and subclasses.

ALGORITHM:

1. Create an abstract class with the objects, constructor to initialise them and the declaration of the print method.
2. Create different subclasses for each shape that extends from the abstract class. Define the print method according to the shape.
3. Create a main class for the program to run.
4. Receive the inputs from the user during the runtime.
5. Compute the Area of the selected shape.
6. Call the method to print the Area.

PROGRAM:

// Abstract Class

package Lab4;

abstract class shape {

double a;

double b;

shape(double a, double b) {

this.a = a;

this.b = b;

}

abstract void printArea();

}

// Rectangle subclass

package Lab4;

public class rectangle extends shape {

public rectangle(double length, double breadth) {

super(length, breadth);

}

@Override

```
void printArea() {  
    System.out.println("Area of Rectangle = " + (a * b));  
}  
  
}  
  
// Triangle subclass  
package Lab4;  
  
public class triangle extends shape {  
    public triangle(double base, double height) {  
        super(base, height);  
    }  
  
    @Override  
    void printArea() {  
        System.out.println("Area of Triangle = " + (a * b / 2));  
    }  
}  
  
// Circle subclass  
package Lab4;  
  
public class circle extends shape {  
    public circle(double radius) {
```

```
        super(radius, 0);  
    }
```

```
    @Override  
    void printArea() {  
        System.out.println("Area of Circle = " + (3.14 * a * a));  
    }  
}
```

```
// Main Class
```

```
package Lab4;
```

```
import java.util.*;
```

```
public class lab4_1 {  
    public static void main(String args[]) {  
        Scanner input = new Scanner(System.in);  
        System.out.println("Hello:");  
        System.out.println("1. Rectangle");  
        System.out.println("2. Triangle");  
        System.out.println("3. Circle");  
        int choice = input.nextInt();  
        switch (choice) {  
            case 1:  
                System.out.println("Enter the length:");  
                double length = input.nextDouble();
```

```
System.out.println("Enter the breadth:");  
double breadth = input.nextDouble();  
rectangle objRectangle = new rectangle(length, breadth);  
objRectangle.printArea();  
break;
```

case 2:

```
System.out.println("Enter the base:");  
double base = input.nextDouble();  
System.out.println("Enter the height:");  
double height = input.nextDouble();  
triangle objTriangle = new triangle(base, height);  
objTriangle.printArea();  
break;
```

case 3:

```
System.out.println("Enter the radius:");  
double radius = input.nextDouble();  
circle objcCircle = new circle(radius);  
objcCircle.printArea();  
break;
```

default:

```
System.out.println("Invalid Input..");  
break;
```

```
}
```

```
}
```

```
}
```

OUTPUT:

```
ashwin_vp@Ubuntu:~$ cd Desktop
ashwin_vp@Ubuntu:~/Desktop$ javac Java/Lab4/*.java
ashwin_vp@Ubuntu:~/Desktop$ cd Java
ashwin_vp@Ubuntu:~/Desktop/Java$ java Lab4.lab4_1
Hello:
1. Rectangle
2. Triangle
3. Circle
1
Enter the length:
10
Enter the breadth:
20
Area of Rectangle = 200.0
ashwin_vp@Ubuntu:~/Desktop/Java$ java Lab4.lab4_1
Hello:
1. Rectangle
2. Triangle
3. Circle
2
Enter the base:
10
Enter the height:
20
Area of Triangle = 100.0
ashwin_vp@Ubuntu:~/Desktop/Java$ java Lab4.lab4_1
Hello:
1. Rectangle
2. Triangle
3. Circle
3
Enter the radius:
10
Area of Circle = 314.0
ashwin_vp@Ubuntu:~/Desktop/Java$ java Lab4.lab4_1
Hello:
1. Rectangle
2. Triangle
3. Circle
4
Invalid Input..
ashwin_vp@Ubuntu:~/Desktop/Java$
```

PROGRAM 2

AIM:

To create a Java program that demonstrates the concept of Overriding.

ALGORITHM:

1. Create an abstract class with the objects, constructor to initialise them and the declaration of the print method.
2. Create different subclasses for each shape that extends from the abstract class. Define the print method according to the shape.
3. Create a main class for the program to run.
4. Receive the inputs from the user during the runtime.
5. Compute the Area for both the shapes.
6. Call the method to print the Area.

PROGRAM:

```
// Abstract Class
```

```
package Lab4;
```

```
abstract class baseclass {
```

```
    double dim1;
```

```
    double dim2;
```

```
    public baseclass(double dim1, double dim2) {
```

```
        this.dim1 = dim1;
```

```
        this.dim2 = dim2;
```

```
    }
```

```

    abstract void area();
}
// Rectangle subclass
package Lab4;
public class rectangleQues2 extends baseclass {
    public rectangleQues2(double length, double breadth) {
        super(length, breadth);
    }
    @Override
    void area() {
        System.out.println("Area of Rectangle = " + dim1 * dim2);
    }
}
// Triangle subclass
package Lab4;
public class triangleQues2 extends baseclass {
    public triangleQues2(double base, double height) {
        super(base, height);
    }
    @Override
    void area() {
        System.out.println("Area of Triangle = " + (dim1 * dim2 / 2));
    }
}

```



```
// Main Class
package Lab4;
import java.util.*;
public class lab4_2 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.println("Enter Dimension 1 : ");
        double dimension1 = input.nextDouble();
        System.out.println("Enter Dimension 2 : ");
        double dimension2 = input.nextDouble();
        System.out.println("Example of Overriding : ");
        rectangleQues2 objRectangle = new rectangleQues2(dimension1,
dimension2);
        triangleQues2 objTriangle = new triangleQues2(dimension1,
dimension2);
        objRectangle.area();
        objTriangle.area();
    }
}
```

OUTPUT:

```
ashwin_vp@Ubuntu:~/Desktop/Java$ java Lab4.lab4_2
Enter Dimension 1 :
10
Enter Dimension 2 :
20
Example of Overriding :
Area of Rectangle = 200.0
Area of Triangle = 100.0
ashwin_vp@Ubuntu:~/Desktop/Java$
```

PROGRAM 3

AIM:

To create a Java program that demonstrates the concept of Overloading.

ALGORITHM:

1. Create a main class for the program to run.
2. Define two methods with the same name, but different number of parameters.
3. Receive the inputs from the user during the runtime.
4. Compute the Area for corresponding shape.
5. Print the Area.

PROGRAM:

```
package Lab4;
```

```
import java.util.*;
```

```
public class lab4_3 {  
    public static double area(double length, double breadth) {  
        return length * breadth;  
    }  
  
    public static double area(double length) {  
        return length * length;  
    }  
}
```

```
}
```

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

```
    Scanner input = new Scanner(System.in);
```

```
    boolean loopController = true;
```

```
    while (loopController) {
```

```
        System.out.println("Hello!");
```

```
        System.out.println("1. Rectangle");
```

```
        System.out.println("2. Square");
```

```
        System.out.println("3. Exit");
```

```
        int choice = input.nextInt();
```

```
        switch (choice) {
```

```
            case 1:
```

```
                System.out.println("Enter Length : ");
```

```
                double length = input.nextDouble();
```

```
                System.out.println("Enter Breadth : ");
```

```
                double breadth = input.nextDouble();
```

```
                System.out.println("Area of Rectangle = " + area(length,  
breadth));
```

```
                break;
```

case 2:

```
System.out.println("Enter Length : ");  
double side = input.nextDouble();  
System.out.println("Area of Square = " + area(side));  
break;
```

case 3:

```
loopController = false;  
System.out.println("Exiting...");  
break;
```

default:

```
System.out.println("Invalid Input..");  
break;
```

```
}
```

```
}
```

```
}
```

```
}
```

OUTPUT:

```
ashwin_vp@Ubuntu:~/Desktop/Java$ java Lab4.lab4_3
Hello!
1. Rectangle
2. Square
3. Exit
1
Enter Length :
10
Enter Breadth :
20
Area of Rectangle = 200.0
Hello!
1. Rectangle
2. Square
3. Exit
2
Enter Length :
10
Area of Square = 100.0
Hello!
1. Rectangle
2. Square
3. Exit
4
Invalid Input..
Hello!
1. Rectangle
2. Square
3. Exit
3
Exiting...
ashwin_vp@Ubuntu:~/Desktop/Java$
```

RESULT:

Thus, Java applications to demonstrate Abstraction and Polymorphism have been created. In Overriding, the same method has different definitions, whereas in Overloading, the number of parameters is different for the same method name.