Practical-6.1 : Write an abstract class shape, which defines the abstract method area. Derive class circle from shape. It has data member radius and implementation for area function. Derive class Triangle from shape. It has data members height, base and implementation for area function. Derive class Square from shape. It has data member side and implementation for area function. In the main class, use dynamic method dispatch in order to call the correct version of the method.

Input:-

```
abstract class Shape {
    abstract void area();
}

class Circle extends Shape {
    double radius;
    Circle(double radius) {
        this.radius = radius;
    }
    void area() {
        System.out.println("Circle Area: " + (Math.PI * radius * radius));
    }
}

class Triangle extends Shape {
    double base, height;
    Triangle(double base, double height) {
        this.base = base;
        this.height = height;
    }
    void area() {
        System.out.println("Triangle Area: " + (0.5 * base * height));
    }
}
class Square extends Shape {
    double side;
    Square(double side) {
        this.side = side;
    }
    void area() {
        System.out.println("Square Area: " + (side * side));
    }
}
```

```
public class ShapeDemo {
    public static void main(String[] args) {
        Shape ref;
        ref = new Circle(5);
        ref.area();
        ref = new Triangle(4, 6);
        ref.area();
        ref = new Square(3);
        ref.area();
    }
}
```

Output:-

```
Circle Area: 78.53981633974483

Triangle Area: 12.0

Square Area: 9.0
```