

SCALA PROGRAMMING

ASSIGNMENT - 5

NAME - KAPAROTU VENKATA SURYA THARANI

USN - 22BTRAD018

BRANCH - AIDE

Ques - Write a Scala program that creates an abstract class Shape with an abstract method area. Implement subclasses Rectangle and Circle that override the area method.

CODE :

```
abstract class Shape {
    def area: Double
}
class Rectangle(width: Double, height: Double) extends Shape {
    override def area: Double = width * height
}
class Circle(radius: Double) extends Shape {
    override def area: Double = math.Pi * radius * radius
}
object ShapeApp {
    def main(args: Array[String]): Unit = {
        val rectangle = new Rectangle(7, 5)
        println(s"Rectangle Area: ${rectangle.area}")
        val circle = new Circle(4.5)
        println(s"Circle Area: ${circle.area}")
    }
}
```

OUTPUT :

Rectangle Area: 35.0

Circle Area: 63.61725123519331



The screenshot shows a Scala IDE interface. On the left, a code editor displays the Scala program for calculating the area of a rectangle and a circle. The code defines an abstract class `Shape` with an abstract method `area`, and two subclasses, `Rectangle` and `Circle`, which override the `area` method. An object `ShapeApp` contains a `main` method that creates instances of `Rectangle` and `Circle` and prints their areas. On the right, the IDE's output console shows the results of running the program: "Rectangle Area: 35.0" and "Circle Area: 63.61725123519331".

```
1 abstract class Shape {
2   def area: Double
3 }
4 class Rectangle(width: Double, height: Double) extends Shape {
5   override def area: Double = width * height
6 }
7 class Circle(radius: Double) extends Shape {
8   override def area: Double = math.Pi * radius * radius
9 }
10 object ShapeApp {
11   def main(args: Array[String]): Unit = {
12     val rectangle = new Rectangle(7, 5)
13     println(s"Rectangle Area: ${rectangle.area}")
14     val circle = new Circle(4.5)
15     println(s"Circle Area: ${circle.area}")
16   }
17 }
```

Output:

Rectangle Area: 35.0
Circle Area: 63.61725123519331