SCALA PROGRAMMING

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
```

Screenshot:

```
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}")
 }
}
```

STDIN

Input for the program (Optional)

Output:

```
Rectangle Area: 35.0
Circle Area: 63.61725123519331
```

When we give character in a single quote it will consider it's ASCII value.

Example code and output:

```
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('A', 5)
 println(s"Rectangle Area: ${rectangle.area}")
 val circle = new Circle(4.5)
 println(s"Circle Area: ${circle.area}")
 }
}
```

STDIN

Input for the program (Optional)

Output:

```
Rectangle Area: 325.0
Circle Area: 63.61725123519331
```