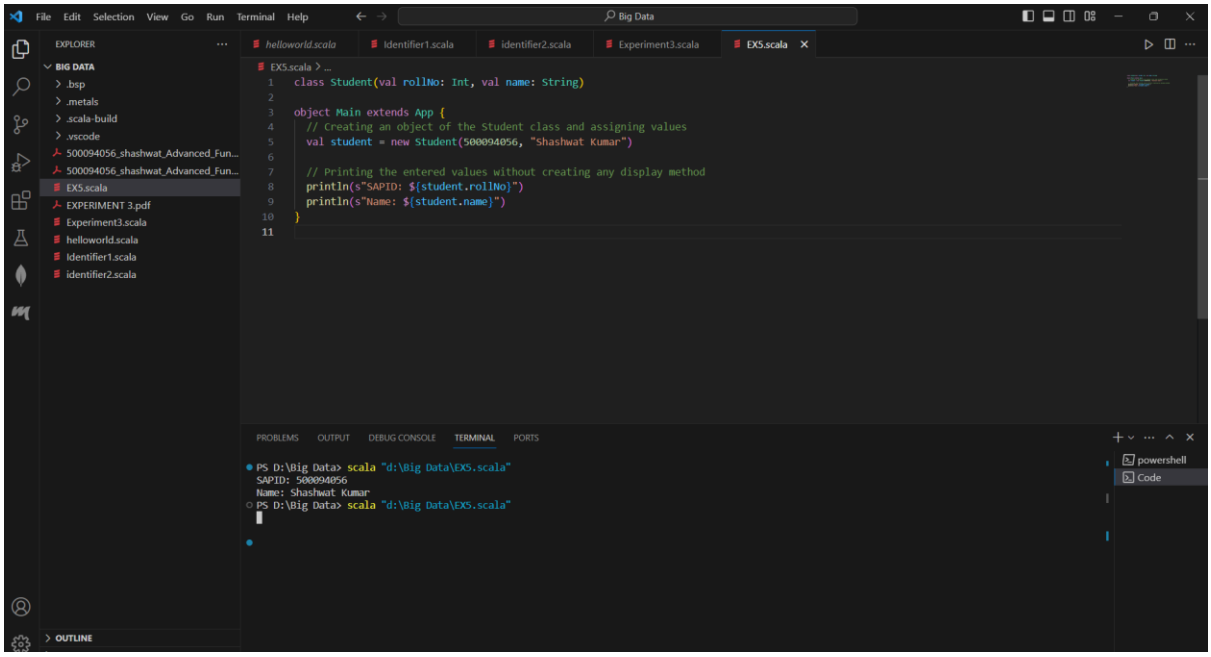
LAB-5

Advanced Functional Thinking

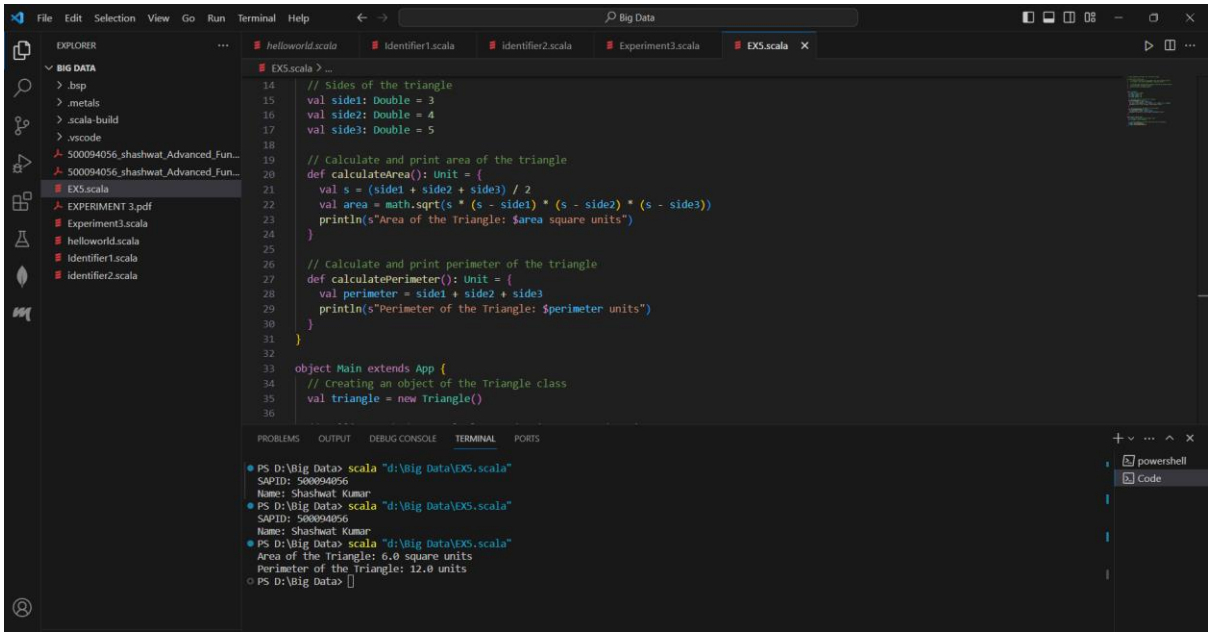
Q.1 Create a class named ‘student’ with string variable and integer variable ‘roll_no’. Assign the value of roll_no and name (your name) by creating an object of the class Student. Print the entered values without creating any display method.

CODE:

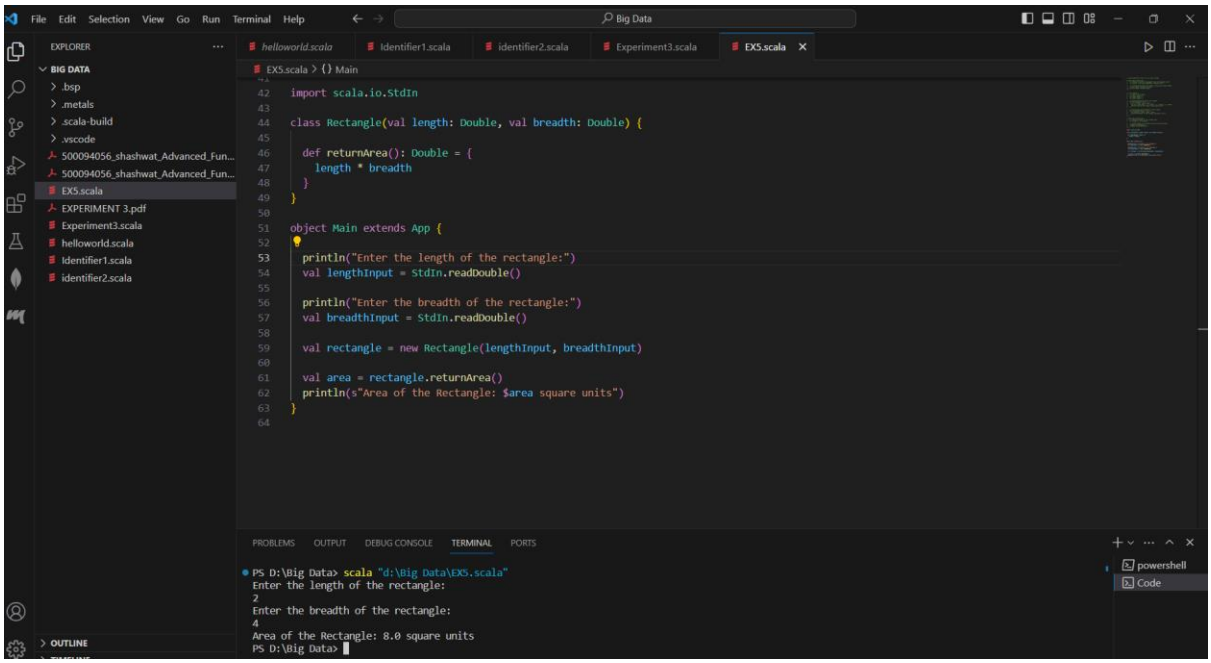


Q2. Write a program to print the area and parameter of a Triangle having sides of 3, 4 and 5 units by creating a class named ‘Triangle’ without any parameter in its constructor.

CODE:



Q3. Write a program to print the area of a rectangle by creating a class named 'Area' taking the values of its length and breadth as parameters of its constructor and having a method named 'returnArea' which returns the area of the rectangle. Length and breadth of rectangles are entered through keyboard.



Q4. Write a program to print the area and parameter of a Triangle having sides of 3, 4 and 5 units by creating a class named 'Triangle' with an Auxiliary constructor having the three sides as its parameters.

Ans-

```
class Triangle(var side1: Double, var side2: Double, var side3: Double) {
  // Auxiliary constructor
  def this() {
    this(3, 4, 5)
  }

  // Method to calculate the perimeter
  def perimeter(): Double = {
    side1 + side2 + side3
  }

  // Method to calculate the area using Heron's formula
  def area(): Double = {
    val s = perimeter() / 2
    math.sqrt(s * (s - side1) * (s - side2) * (s - side3))
  }
}

object Main {
  def main(args: Array[String]): Unit = {
    val triangle = new Triangle()
    val trianglePerimeter = triangle.perimeter()
    val triangleArea = triangle.area()
    println(s"Perimeter of the triangle: $trianglePerimeter units")
    println(s"Area of the triangle: $triangleArea square units")
  }
}
```

EX5.scala

```scala
class Triangle(side1: Double, side2: Double, side3: Double) {
  // Auxiliary constructor
  def this() {
    this(3, 4, 5)
  }

  // Method to calculate the perimeter
  def perimeter(): Double = {
    side1 + side2 + side3
  }

  // Method to calculate the area using Heron's formula
  def area(): Double = {
    val s = perimeter() / 2
    math.sqrt(s * (s - side1) * (s - side2) * (s - side3))
  }
}

object Main {
  def main(args: Array[String]): Unit = {
    val triangle = new Triangle()
    val trianglePerimeter = triangle.perimeter()
    val triangleArea = triangle.area()
    println(s"Perimeter of the triangle: $trianglePerimeter units")
    println(s"Area of the triangle: $triangleArea square units")
  }
}

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\Big Data>
```

Q5. Create a base class Shape with a method draw. Then, create subclasses Circle, Rectangle, and Triangle that override the draw method to draw their respective shapes.

Ans-

EX5.scala > {} Main

```scala
abstract class Shape {
}
class Circle extends Shape {
  ↑ draw
override def draw(): Unit = println("Drawing a circle")
}
class Rectangle extends Shape {
  ↑ draw
override def draw(): Unit = println("Drawing a rectangle")
}
class Triangle extends Shape {
  ↑ draw
override def draw(): Unit = println("Drawing a triangle")
}
object Main {
def main(args: Array[String]): Unit = {
val circle = new Circle()
val rectangle = new Rectangle()
val triangle = new Triangle()
println("Drawing shapes:")
circle.draw()
rectangle.draw()
triangle.draw()
}
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\Big Data> scala "d:\Big Data\EX5.scala"
Drawing shapes:
Drawing a circle
Drawing a rectangle
Drawing a triangle
PS D:\Big Data>
```