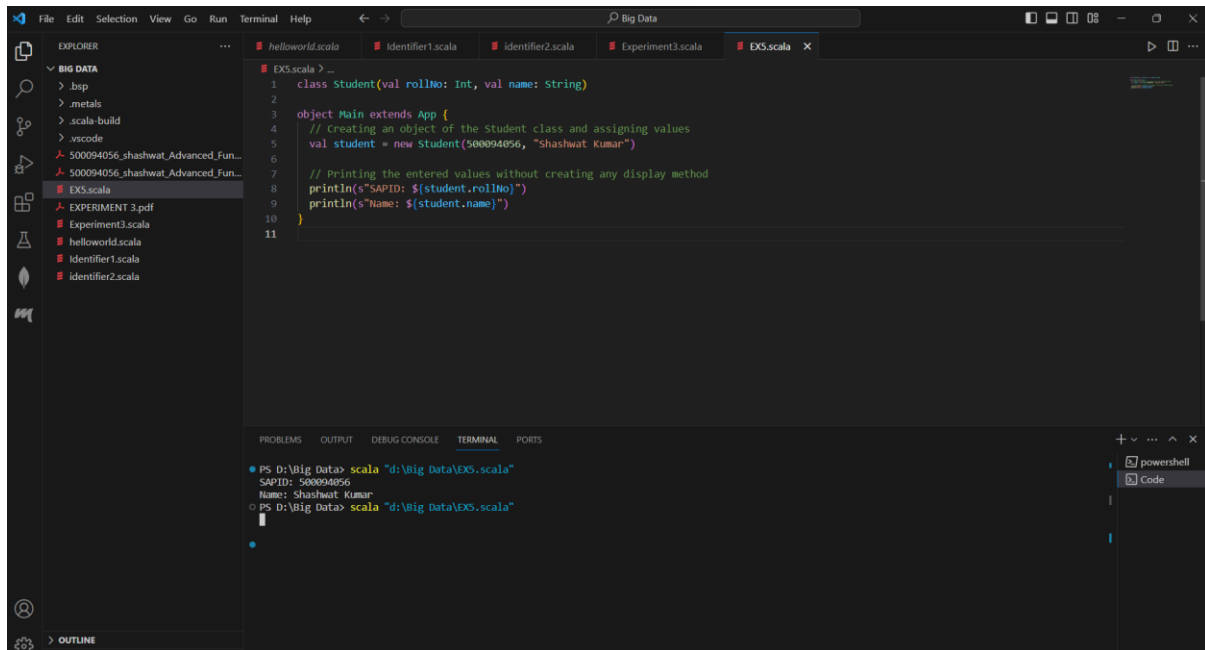


LAB-5

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:



The screenshot shows the Visual Studio Code editor with a Scala file named `EX5.scala` open. The code defines a `Student` class with two parameters: `rollNo: Int` and `name: String`. An object `Main` extends `App` and contains a `main` method. In the `main` method, a `Student` object is created with the roll number `500094056` and the name `"Shashwat Kumar"`. The program then prints the `rollNo` and `name` attributes of the `student` object. The terminal at the bottom shows the command `scala "d:\Big Data\EX5.scala"` being executed, and the output displays the roll number and name.

```
1 class Student(val rollNo: Int, val name: String)
2
3 object Main extends App {
4   // Creating an object of the Student class and assigning values
5   val student = new Student(500094056, "Shashwat Kumar")
6
7   // Printing the entered values without creating any display method
8   println(s"SAPID: ${student.rollNo}")
9   println(s"Name: ${student.name}")
10
11 }
```

Terminal Output:

```
PS D:\Big Data> scala "d:\Big Data\EX5.scala"
SAPID: 500094056
Name: Shashwat Kumar
PS D:\Big Data> scala "d:\Big Data\EX5.scala"
```

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:

```

14 // Sides of the triangle
15 val side1: Double = 3
16 val side2: Double = 4
17 val side3: Double = 5
18
19 // Calculate and print area of the triangle
20 def calculateArea(): Unit = {
21   val s = (side1 + side2 + side3) / 2
22   val area = math.sqrt(s * (s - side1) * (s - side2) * (s - side3))
23   println(s"Area of the Triangle: $area square units")
24 }
25
26 // Calculate and print perimeter of the triangle
27 def calculatePerimeter(): Unit = {
28   val perimeter = side1 + side2 + side3
29   println(s"Perimeter of the Triangle: $perimeter units")
30 }
31
32
33 object Main extends App {
34   // Creating an object of the Triangle class
35   val triangle = new Triangle()
36 }

```

```

PS D:\Big Data> scala "d:\Big Data\EX5.scala"
SAPID: 500094056
Name: Shashwat Kumar
PS D:\Big Data> scala "d:\Big Data\EX5.scala"
SAPID: 500094056
Name: Shashwat Kumar
PS D:\Big Data> scala "d:\Big Data\EX5.scala"
Area of the Triangle: 6.0 square units
Perimeter of the Triangle: 12.0 units
PS D:\Big Data>

```

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.

```

42 import scala.io.StdIn
43
44 class Rectangle(val length: Double, val breadth: Double) {
45
46   def returnArea(): Double = {
47     length * breadth
48   }
49 }
50
51 object Main extends App {
52   println("Enter the length of the rectangle:")
53   val lengthInput = StdIn.readDouble()
54
55   println("Enter the breadth of the rectangle:")
56   val breadthInput = StdIn.readDouble()
57
58   val rectangle = new Rectangle(lengthInput, breadthInput)
59
60   val area = rectangle.returnArea()
61   println(s"Area of the Rectangle: $area square units")
62 }
63
64

```

```

PS D:\Big Data> scala "d:\Big Data\EX5.scala"
Enter the length of the rectangle:
2
Enter the breadth of the rectangle:
4
Area of the Rectangle: 8.0 square units
PS D:\Big Data>

```

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
1 class Triangle(side1: Double, side2: Double, side3: Double) {
2     // Auxiliary constructor
3     def this() {
4         this(3, 4, 5)
5     }
6
7     // Method to calculate the perimeter
8     def perimeter(): Double = {
9         side1 + side2 + side3
10    }
11
12    // Method to calculate the area using Heron's formula
13    def area(): Double = {
14        val s = perimeter() / 2
15        math.sqrt(s * (s - side1) * (s - side2) * (s - side3))
16    }
17 }
18
19 object Main {
20     def main(args: Array[String]): Unit = {
21         val triangle = new Triangle()
22         val trianglePerimeter = triangle.perimeter()
23         val triangleArea = triangle.area()
24         println(s"Perimeter of the triangle: $trianglePerimeter units")
25         println(s"Area of the triangle: $triangleArea square units")
26     }
27 }
28
```

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
1 abstract class Shape {
2     }
3
4 class Circle extends Shape {
5     ↑ draw
6     override def draw(): Unit = println("Drawing a circle")
7 }
8 class Rectangle extends Shape {
9     ↑ draw
10    override def draw(): Unit = println("Drawing a rectangle")
11 }
12 class Triangle extends Shape {
13     ↑ draw
14    override def draw(): Unit = println("Drawing a triangle")
15 }
16
17 object Main {
18     def main(args: Array[String]): Unit = {
19         val circle = new Circle()
20         val rectangle = new Rectangle()
21         val triangle = new Triangle()
22         println("Drawing shapes:")
23         circle.draw()
24         rectangle.draw()
25         triangle.draw()
26     }
27 }
28
```

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>