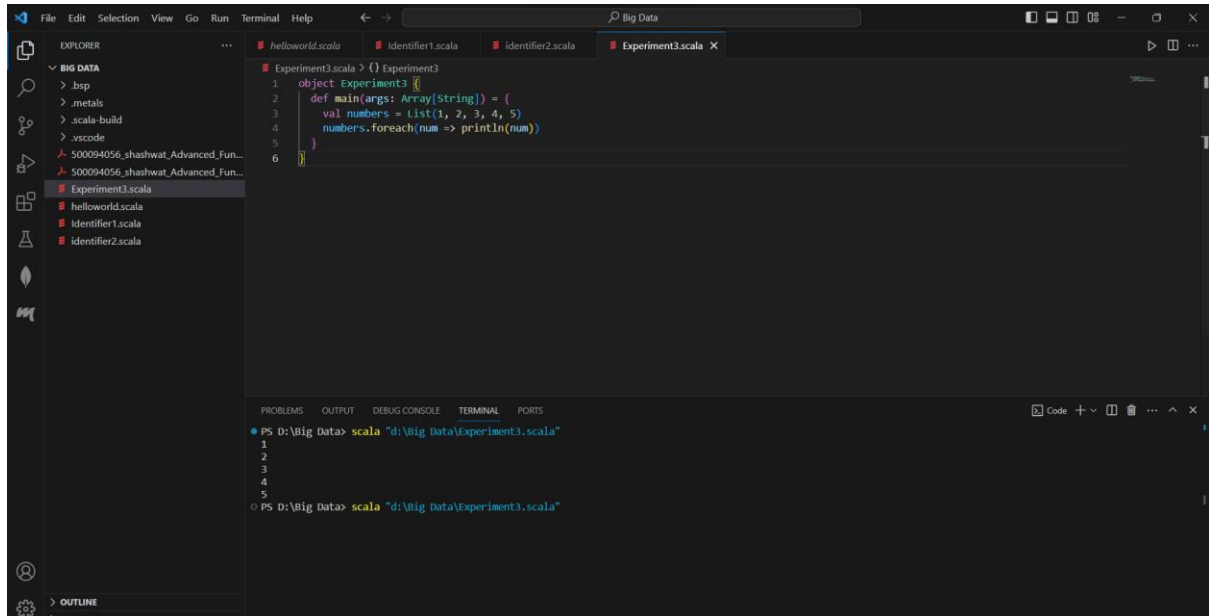


EXPERIMENT 3

Shashwat kumar

500094056

Q1. Scala program to implement the foreach loop on a list of numbers.

A screenshot of the Visual Studio Code editor interface. The Explorer sidebar on the left shows a project named 'BIG DATA' with files like '.bsp', '.metals', '.scala-build', '.vscode', and 'Experiment3.scala'. The main editor window displays the code for 'Experiment3.scala'. The code defines an object 'Experiment3' with a 'main' method that creates a list of numbers (1, 2, 3, 4, 5) and uses the 'foreach' method to print each number. The bottom panel shows the 'TERMINAL' tab with the command 'scala "d:\Big Data\Experiment3.scala"' and its output, which lists the numbers 1 through 5 on separate lines.

```
1 object Experiment3 {  
2   def main(args: Array[String]) = {  
3     val numbers = List(1, 2, 3, 4, 5)  
4     numbers.foreach(num => println(num))  
5   }  
6 }
```

```
PS D:\Big Data> scala "d:\Big Data\Experiment3.scala"  
1  
2  
3  
4  
5  
PS D:\Big Data> scala "d:\Big Data\Experiment3.scala"
```

```
object ForEach {  
  
  def main(args: Array[String]) = {  
  
    val numbers = List(1, 2, 3, 4, 5)  
  
    numbers.foreach(num => println(num))  
  
  }  
  
}
```

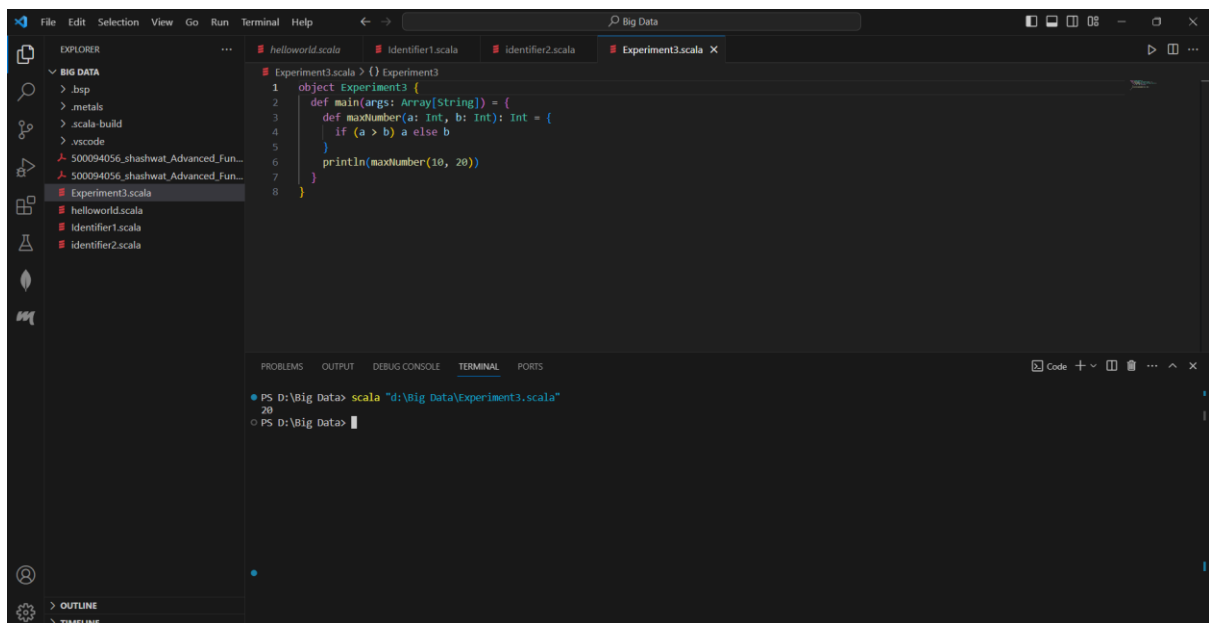
Q2. Scala program to create a user define function to return largest number among two numbers entered by user.

```
object MaxNumber {  
  
  def main(args: Array[String]) = {  
  
    def maxNumber(a: Int, b: Int): Int = {  
  
      if (a > b) a else b  
  
    }  
  
  }  
  
}
```

```

println(maxNumber(10, 20))
}
}

```

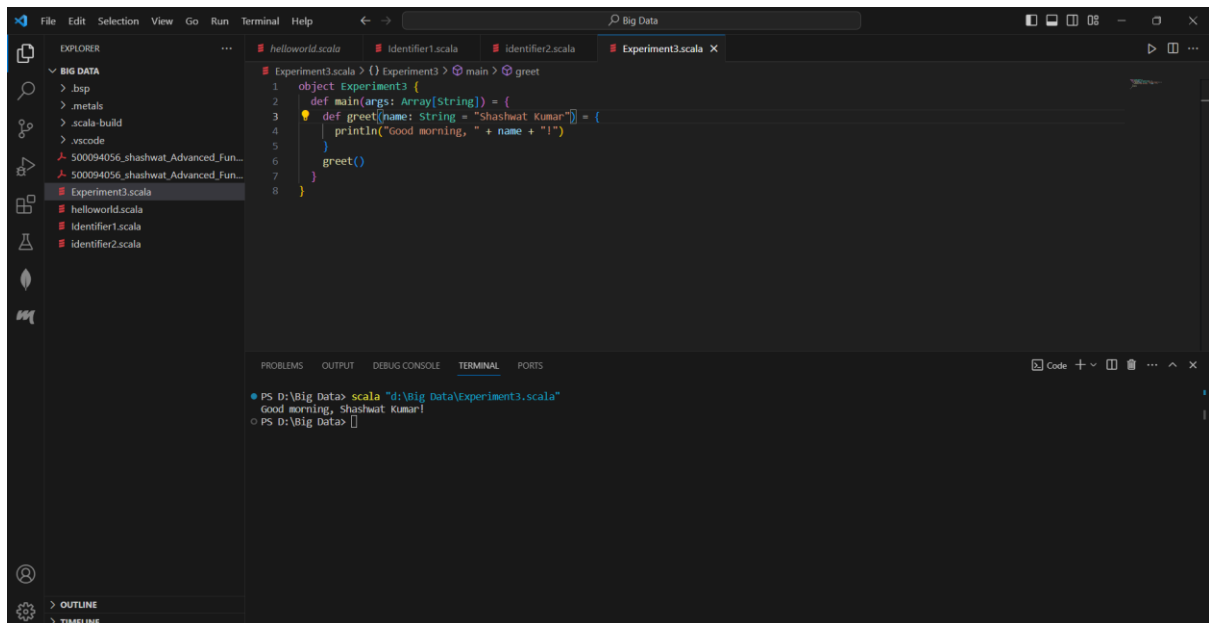


Q3. Scala code to create a function with default arguments. Wish good morning to the person.

```

object Greet {
  def main(args: Array[String]) = {
    def greet(name: String = "Shashwat") = {
      println("Good morning, " + name + "!")
    }
    greet()
  }
}

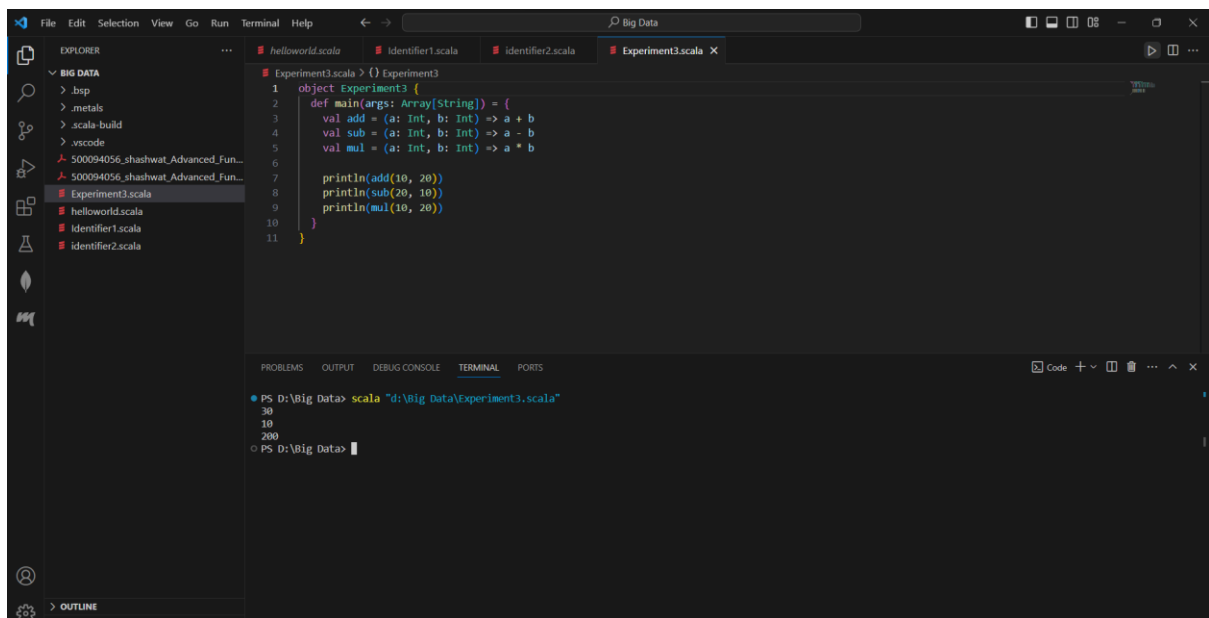
```



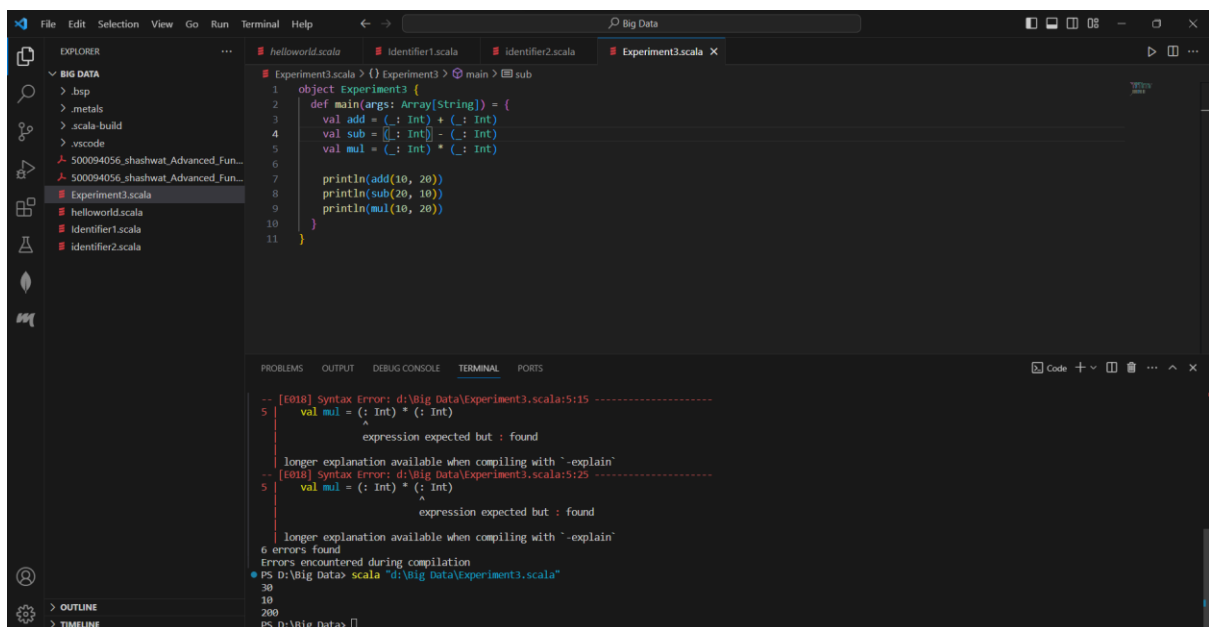
Q4. Scala code to create anonymous functions for add, sub, and mul with => operator.

```
object AnonymousFunctions {
  def main(args: Array[String]) = {
    val add = (a: Int, b: Int) => a + b
    val sub = (a: Int, b: Int) => a - b
    val mul = (a: Int, b: Int) => a * b

    println(add(10, 20))
    println(sub(20, 10))
    println(mul(10, 20))
  }
}
```



Q5. Scala code to create anonymous functions for add, sub, and mul with _ operator.



```

object Experiment3 {
  def main(args: Array[String]) = {
    val add = (_: Int) + (_: Int)
    val sub = (_: Int) - (_: Int)
    val mul = (_: Int) * (_: Int)

```

```
println(add(10, 20))  
println(sub(20, 10))  
println(mul(10, 20))  
}  
}
```