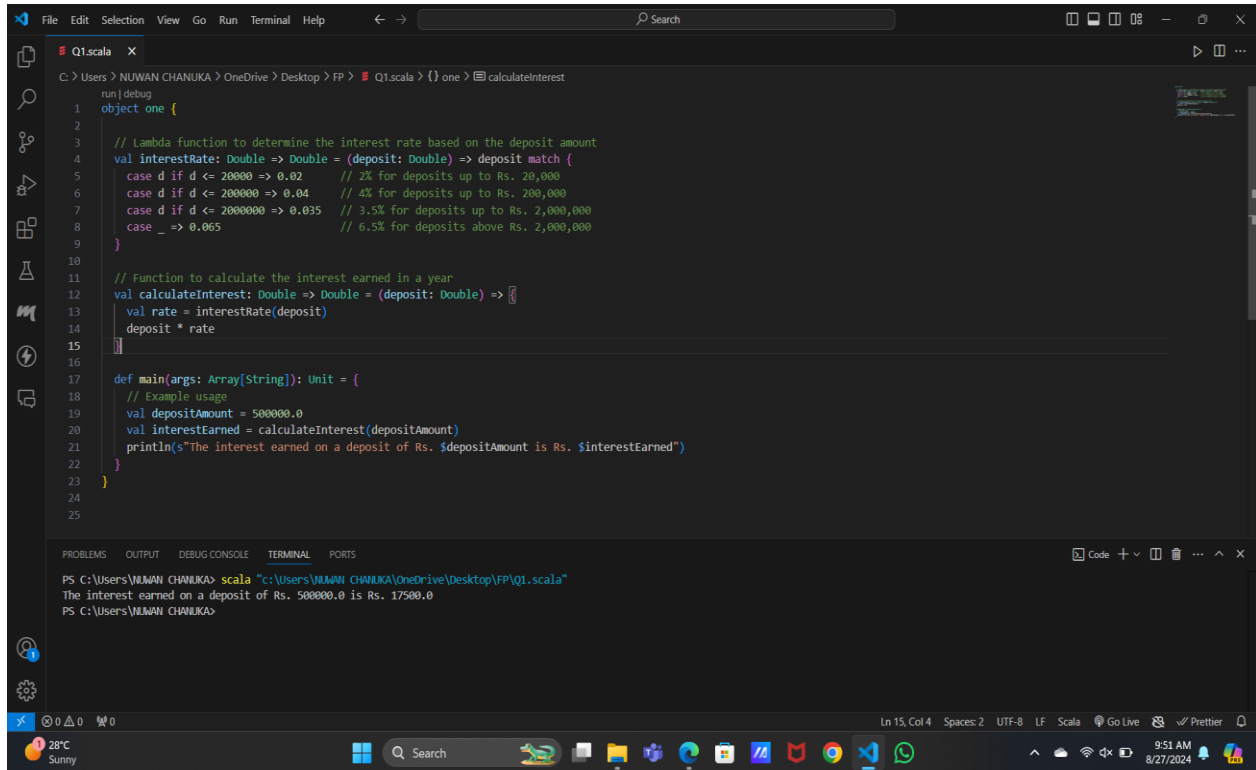


# 22000992 -Practical 09

Q 01



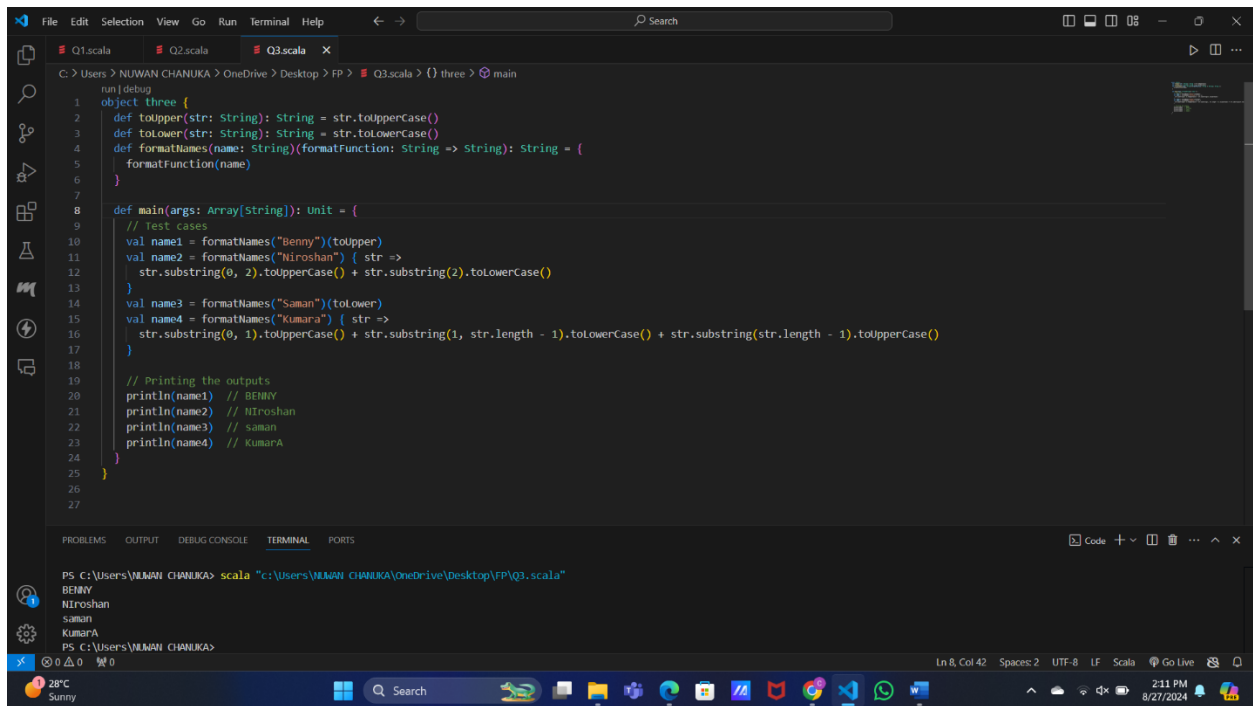
The screenshot shows an IDE with a Scala file named `Q1.scala`. The code defines a `one` object with a lambda function `interestRate` and a function `calculateInterest`. The `main` function uses these to calculate the interest on a deposit of 500,000.0.

```
1  object one {  
2  
3      // Lambda function to determine the interest rate based on the deposit amount  
4      val interestRate: Double => Double = (deposit: Double) => deposit match {  
5          case d if d <= 20000 => 0.02 // 2% for deposits up to Rs. 20,000  
6          case d if d <= 200000 => 0.04 // 4% for deposits up to Rs. 200,000  
7          case d if d <= 2000000 => 0.035 // 3.5% for deposits up to Rs. 2,000,000  
8          case _ => 0.065 // 6.5% for deposits above Rs. 2,000,000  
9      }  
10  
11      // Function to calculate the interest earned in a year  
12      val calculateInterest: Double => Double = (deposit: Double) => {  
13          val rate = interestRate(deposit)  
14          deposit * rate  
15      }  
16  
17      def main(args: Array[String]): Unit = {  
18          // Example usage  
19          val depositAmount = 500000.0  
20          val interestEarned = calculateInterest(depositAmount)  
21          println(s"The interest earned on a deposit of Rs. $depositAmount is Rs. $interestEarned")  
22      }  
23  }  
24  
25
```

The terminal output shows the command to run the Scala file and the resulting interest calculation:

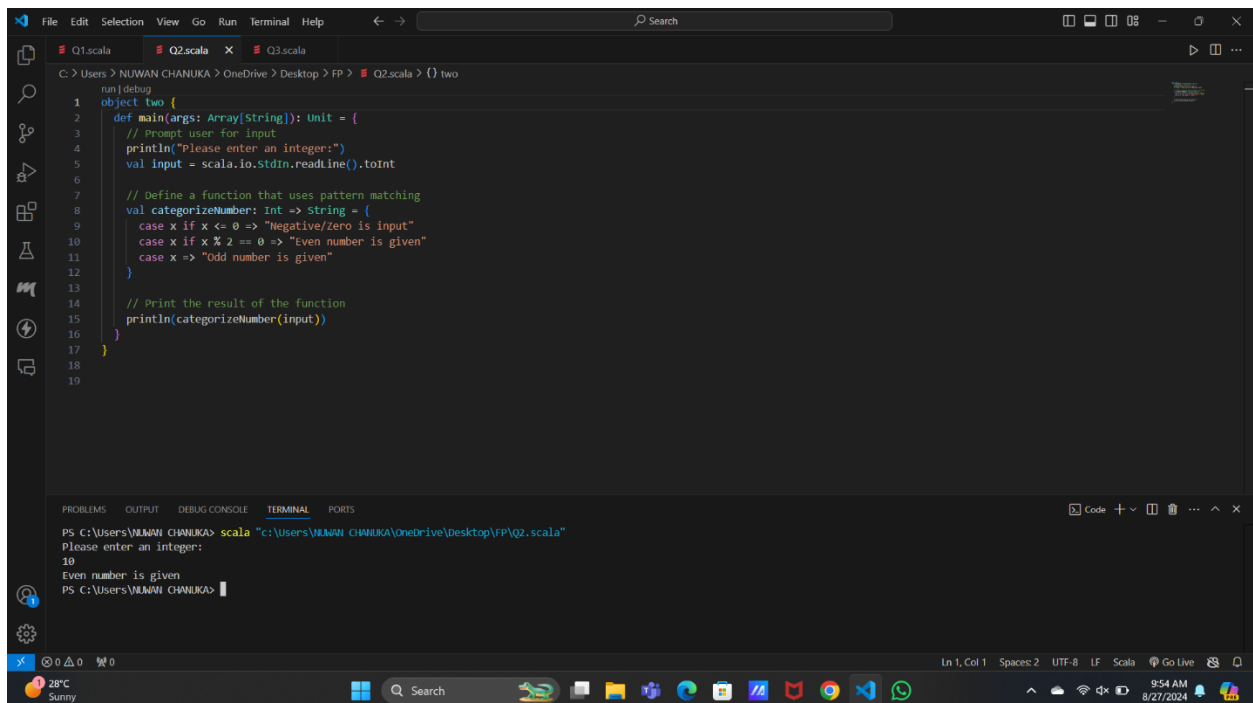
```
PS C:\Users\WILMAN CHANUKA> scala "c:\Users\WILMAN CHANUKA\OneDrive\Desktop\FP\Q1.scala"  
The interest earned on a deposit of Rs. 500000.0 is Rs. 17500.0  
PS C:\Users\WILMAN CHANUKA>
```

Q 03



```
File Edit Selection View Go Run Terminal Help
C:\Users\NUNWAN CHANUKA> OneDrive > Desktop > FP > Q3.scala > {} three > main
run | debug
1 object three {
2   def toUpper(str: String): String = str.toUpperCase()
3   def toLower(str: String): String = str.toLowerCase()
4   def formatNames(name: String)(formatFunction: String => String): String = {
5     formatFunction(name)
6   }
7
8   def main(args: Array[String]): Unit = {
9     // Test cases
10    val name1 = formatNames("Benny")(toUpper)
11    val name2 = formatNames("Niroshan") { str =>
12      str.substring(0, 2).toUpperCase() + str.substring(2).toLowerCase()
13    }
14    val name3 = formatNames("Saman")(toLower)
15    val name4 = formatNames("Kumara") { str =>
16      str.substring(0, 1).toUpperCase() + str.substring(1, str.length - 1).toLowerCase() + str.substring(str.length - 1).toUpperCase()
17    }
18
19    // Printing the outputs
20    println(name1) // BENNY
21    println(name2) // NIROSHAN
22    println(name3) // saman
23    println(name4) // Kumara
24  }
25 }
26
27
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\NUNWAN CHANUKA> scala "c:\Users\NUNWAN CHANUKA\OneDrive\Desktop\FP\Q3.scala"
BENNY
NIROSHAN
saman
Kumara
PS C:\Users\NUNWAN CHANUKA>
```

Q02



```
File Edit Selection View Go Run Terminal Help
C:\Users\NUNWAN CHANUKA> OneDrive > Desktop > FP > Q2.scala > {} two
run | debug
1 object two {
2   def main(args: Array[String]): Unit = {
3     // Prompt user for input
4     println("Please enter an integer:")
5     val input = scala.io.StdIn.readLine().toInt
6
7     // Define a function that uses pattern matching
8     val categorizeNumber: Int => String = {
9       case x if x <= 0 => "Negative/zero is input"
10      case x if x % 2 == 0 => "Even number is given"
11      case x => "Odd number is given"
12    }
13
14    // Print the result of the function
15    println(categorizeNumber(input))
16  }
17 }
18
19
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\NUNWAN CHANUKA> scala "c:\Users\NUNWAN CHANUKA\OneDrive\Desktop\FP\Q2.scala"
Please enter an integer:
10
Even number is given
PS C:\Users\NUNWAN CHANUKA>
```