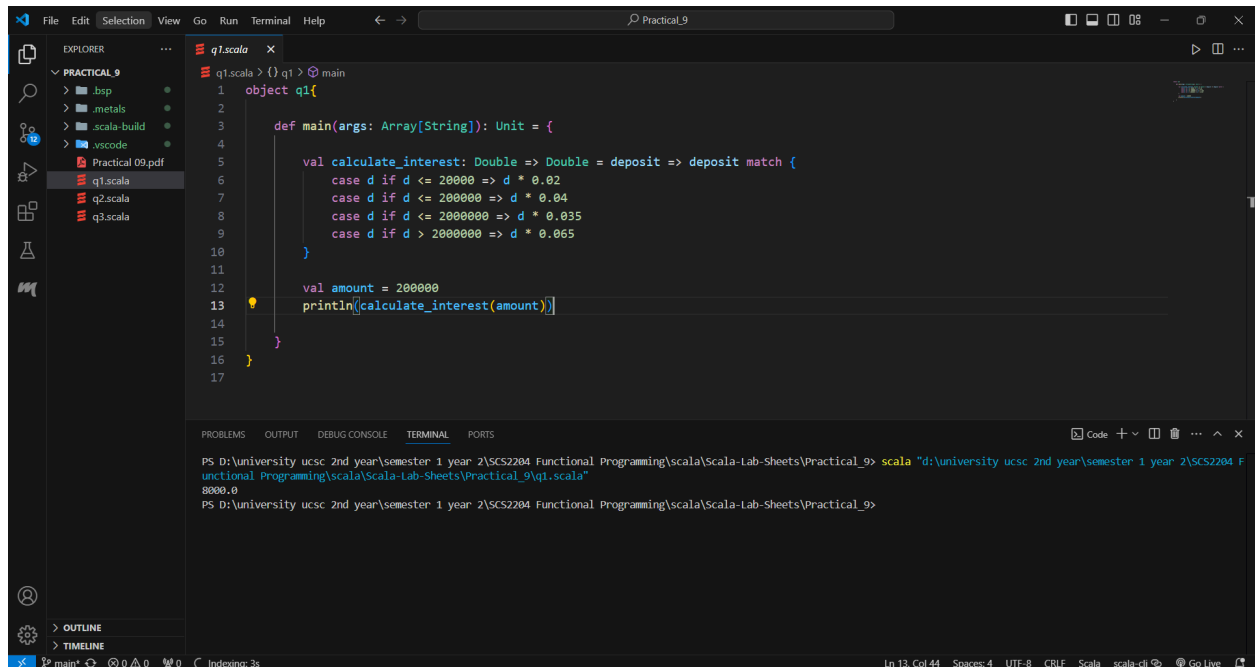


Practical 9

22000518
M.A.Ekanayake

Q1.

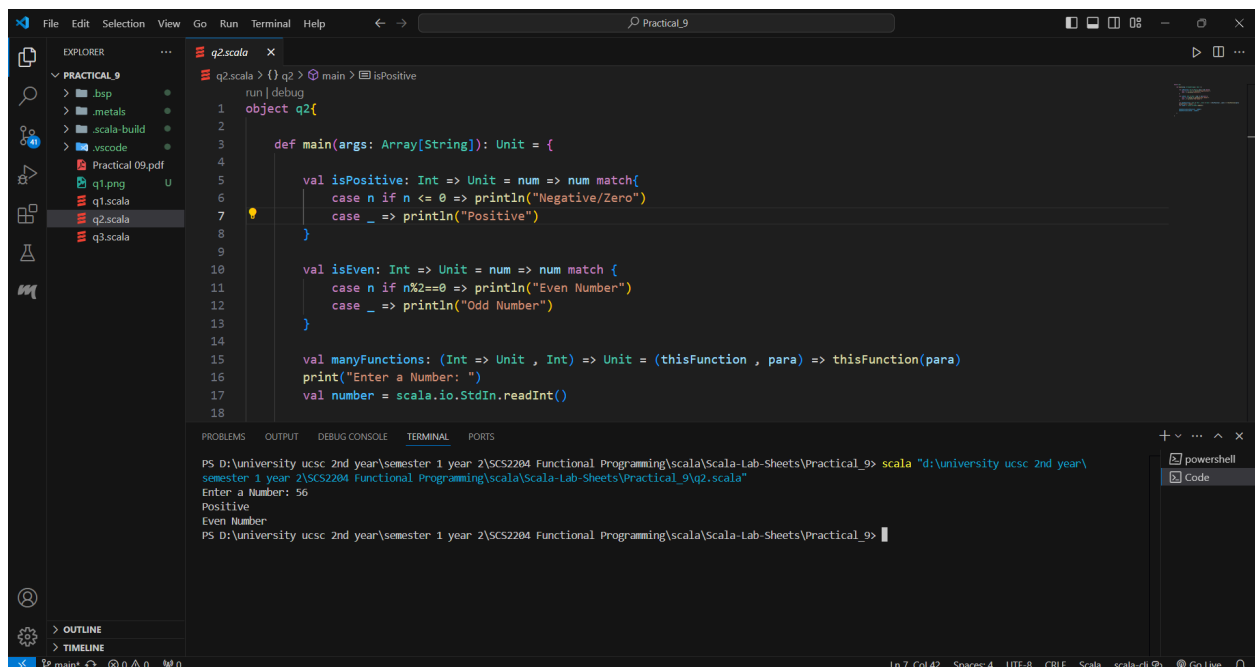


The screenshot shows a VS Code editor with a Scala file named `q1.scala`. The code defines an object `q1` with a `main` function. Inside `main`, a `calculate_interest` function is defined using a `match` expression to calculate interest based on the deposit amount. The deposit amount is set to 200000, and the result is printed.

```
object q1 {  
  def main(args: Array[String]): Unit = {  
    val calculate_interest: Double => Double = deposit => deposit match {  
      case d if d <= 20000 => d * 0.02  
      case d if d <= 200000 => d * 0.04  
      case d if d <= 2000000 => d * 0.035  
      case d if d > 2000000 => d * 0.065  
    }  
    val amount = 200000  
    println(calculate_interest(amount))  
  }  
}
```

The terminal output shows the command `scala "d:\university ucsc 2nd year\semester 1 year 2\SCS2204 Functional Programming\scala\Scala-Lab-Sheets\Practical_9\q1.scala"` and the output `8000.0`.

Q2.

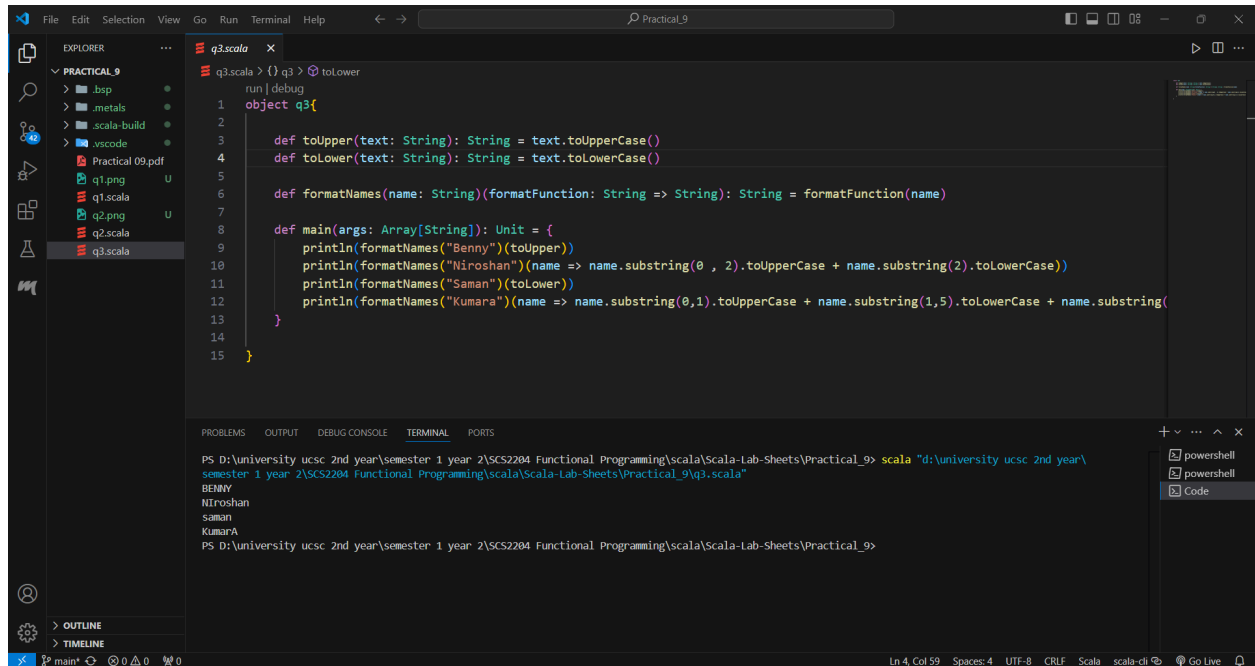


The screenshot shows a VS Code editor with a Scala file named `q2.scala`. The code defines an object `q2` with a `main` function. Inside `main`, two functions are defined: `isPositive` and `isEven`, both using `match` expressions. The `main` function prompts the user to enter a number, reads the input, and prints the results of the `isPositive` and `isEven` functions.

```
object q2 {  
  def main(args: Array[String]): Unit = {  
    val isPositive: Int => Unit = num => num match {  
      case n if n <= 0 => println("Negative/Zero")  
      case _ => println("Positive")  
    }  
    val isEven: Int => Unit = num => num match {  
      case n if n%2==0 => println("Even Number")  
      case _ => println("Odd Number")  
    }  
    val manyFunctions: (Int => Unit, Int) => Unit = (thisFunction, para) => thisFunction(para)  
    print("Enter a Number: ")  
    val number = scala.io.StdIn.readInt()  
    manyFunctions(isPositive, number)  
    manyFunctions(isEven, number)  
  }  
}
```

The terminal output shows the command `scala "d:\university ucsc 2nd year\semester 1 year 2\SCS2204 Functional Programming\scala\Scala-Lab-Sheets\Practical_9\q2.scala"` and the user input `56`. The output shows `Positive` and `Even Number`.

Q3.



The image shows a Visual Studio Code editor window with a Scala file named `q3.scala` open. The file contains a Scala program that defines a `q3` object with methods `toUpper`, `toLower`, and `formatNames`. The `main` function calls `formatNames` with the names "Benny", "Niroshan", "Saman", and "Kumara". The terminal at the bottom shows the command `scala "d:\university ucsc 2nd year\semester 1 year 2\SCS2204 Functional Programming\scala\Scala-Lab-Sheets\Practical_9\q3.scala"` being executed, and the output of the program is displayed.

```
q3.scala > {} q3 > toLower
run | debug
1  object q3{
2
3
4      def toUpper(text: String): String = text.toUpperCase()
5
6      def toLower(text: String): String = text.toLowerCase()
7
8      def formatNames(name: String)(formatFunction: String => String): String = formatFunction(name)
9
10     def main(args: Array[String]): Unit = {
11         println(formatNames("Benny")(toUpper))
12         println(formatNames("Niroshan")(name => name.substring(0, 2).toUpperCase + name.substring(2).toLowerCase))
13         println(formatNames("Saman")(toLower))
14         println(formatNames("Kumara")(name => name.substring(0,1).toUpperCase + name.substring(1,5).toLowerCase + name.substring(5)))
15     }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\university ucsc 2nd year\semester 1 year 2\SCS2204 Functional Programming\scala\Scala-Lab-Sheets\Practical_9> scala "d:\university ucsc 2nd year\semester 1 year 2\SCS2204 Functional Programming\scala\Scala-Lab-Sheets\Practical_9\q3.scala"

BENNY
NIROSHAN
SAMAN
KUMARA

PS D:\university ucsc 2nd year\semester 1 year 2\SCS2204 Functional Programming\scala\Scala-Lab-Sheets\Practical_9>

Ln 4, Col 59 Spaces: 4 UTF-8 CRLF Scala scala-cli Go Live