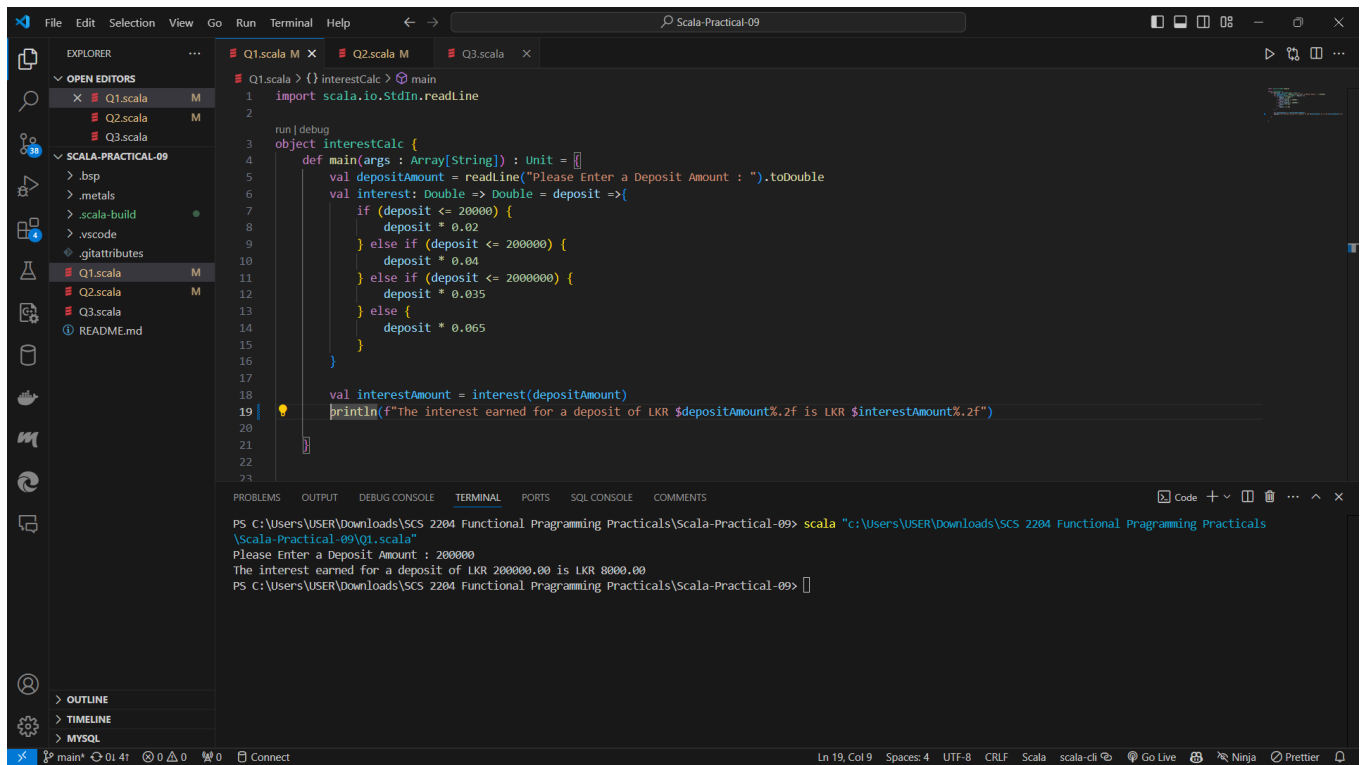


22000862

R.K.J.P.Kashmira

Scala Practical - 09

Q1



The screenshot shows the Visual Studio Code editor with a Scala file named `Q1.scala` open. The code defines an `interestCalc` object with a `main` method that reads a deposit amount from the user and calculates the interest based on three tiers: 0.02 for deposits up to 20,000, 0.04 for deposits up to 200,000, and 0.035 for deposits up to 2,000,000. The interest is then printed out.

```
1 import scala.io.StdIn.readLine
2
3 object interestCalc {
4   def main(args : Array[String]) : Unit = {
5     val depositAmount = readLine("Please Enter a Deposit Amount : ").toDouble
6     val interest: Double => Double = deposit => {
7       if (deposit <= 20000) {
8         deposit * 0.02
9       } else if (deposit <= 200000) {
10        deposit * 0.04
11      } else if (deposit <= 2000000) {
12        deposit * 0.035
13      } else {
14        deposit * 0.065
15      }
16    }
17
18    val interestAmount = interest(depositAmount)
19    println(f"The interest earned for a deposit of LKR $depositAmount%.2f is LKR $interestAmount%.2f")
20  }
21 }
22
23
```

The terminal output shows the execution of the program:

```
PS C:\Users\USER\Downloads\SCS 2204 Functional Programming Practicals\Scala-Practical-09> scala "c:\Users\USER\Downloads\SCS 2204 Functional Programming Practicals\Scala-Practical-09\Q1.scala"
Please Enter a Deposit Amount : 200000
The interest earned for a deposit of LKR 200000.00 is LKR 8000.00
PS C:\Users\USER\Downloads\SCS 2204 Functional Programming Practicals\Scala-Practical-09>
```

Q2

The screenshot shows a VS Code editor with a Scala file named `Q2.scala`. The code defines a `NumClassifier` object with a `main` function that reads a number from the command line and classifies it as even, odd, or negative/zero. The terminal shows the execution of the program with inputs 34, 0, and 3, resulting in the outputs "Even number is given", "Negative/Zero is input", and "Odd number is given" respectively.

```
1 import scala.io.StdIn.readLine
2
3 run | debug
4 object NumClassifier {
5   def main(args: Array[String]): Unit = {
6     val input = readLine("Please Enter a Number : ").toInt
7
8     val classifyNum: Int => String = {
9       case n if n <= 0 => "Negative/Zero is input"
10      case n if n % 2 == 0 => "Even number is given"
11      case _ => "Odd number is given"
12    }
13
14    println(classifyNum(input))
15  }
16 }
```

Terminal Output:

```
PS C:\Users\USER\Downloads\SCS 2204 Functional Programming Practicals\Scala-Practical-09> scala "c:\Users\USER\Downloads\SCS 2204 Functional Programming Practicals\Scala-Practical-09\Q2.scala"
Please Enter a Number : 34
Even number is given
PS C:\Users\USER\Downloads\SCS 2204 Functional Programming Practicals\Scala-Practical-09> scala "c:\Users\USER\Downloads\SCS 2204 Functional Programming Practicals\Scala-Practical-09\Q2.scala"
Please Enter a Number : 0
Negative/Zero is input
PS C:\Users\USER\Downloads\SCS 2204 Functional Programming Practicals\Scala-Practical-09> scala "c:\Users\USER\Downloads\SCS 2204 Functional Programming Practicals\Scala-Practical-09\Q2.scala"
Please Enter a Number : 3
Odd number is given
PS C:\Users\USER\Downloads\SCS 2204 Functional Programming Practicals\Scala-Practical-09>
```

Q3

The screenshot shows a VS Code editor with a Scala file named `Q3.scala`. The code defines a `StringFormatter` object with methods `toUpper`, `toLower`, and `formatNames`. The `main` function uses these methods to format the names "Benny", "Niroshan", "Saman", and "Kumara". The terminal shows the execution of the program, displaying the formatted names in uppercase, lowercase, and a combination of both.

```
1 run | debug
2 object StringFormatter {
3   def toUpper(s: String): String = s.toUpperCase
4   def toLower(s: String): String = s.toLowerCase
5
6   def formatNames(name: String)(formatFunc: String => String): String = {
7     formatFunc(name)
8   }
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).toUpperCase))
15  }
16 }
```

Terminal Output:

```
PS C:\Users\USER\Downloads\SCS 2204 Functional Programming Practicals\Scala-Practical-09> scala "c:\Users\USER\Downloads\SCS 2204 Functional Programming Practicals\Scala-Practical-09\Q3.scala"
BENNY
NIROSHAN
saman
KUMARA
PS C:\Users\USER\Downloads\SCS 2204 Functional Programming Practicals\Scala-Practical-09>
```

