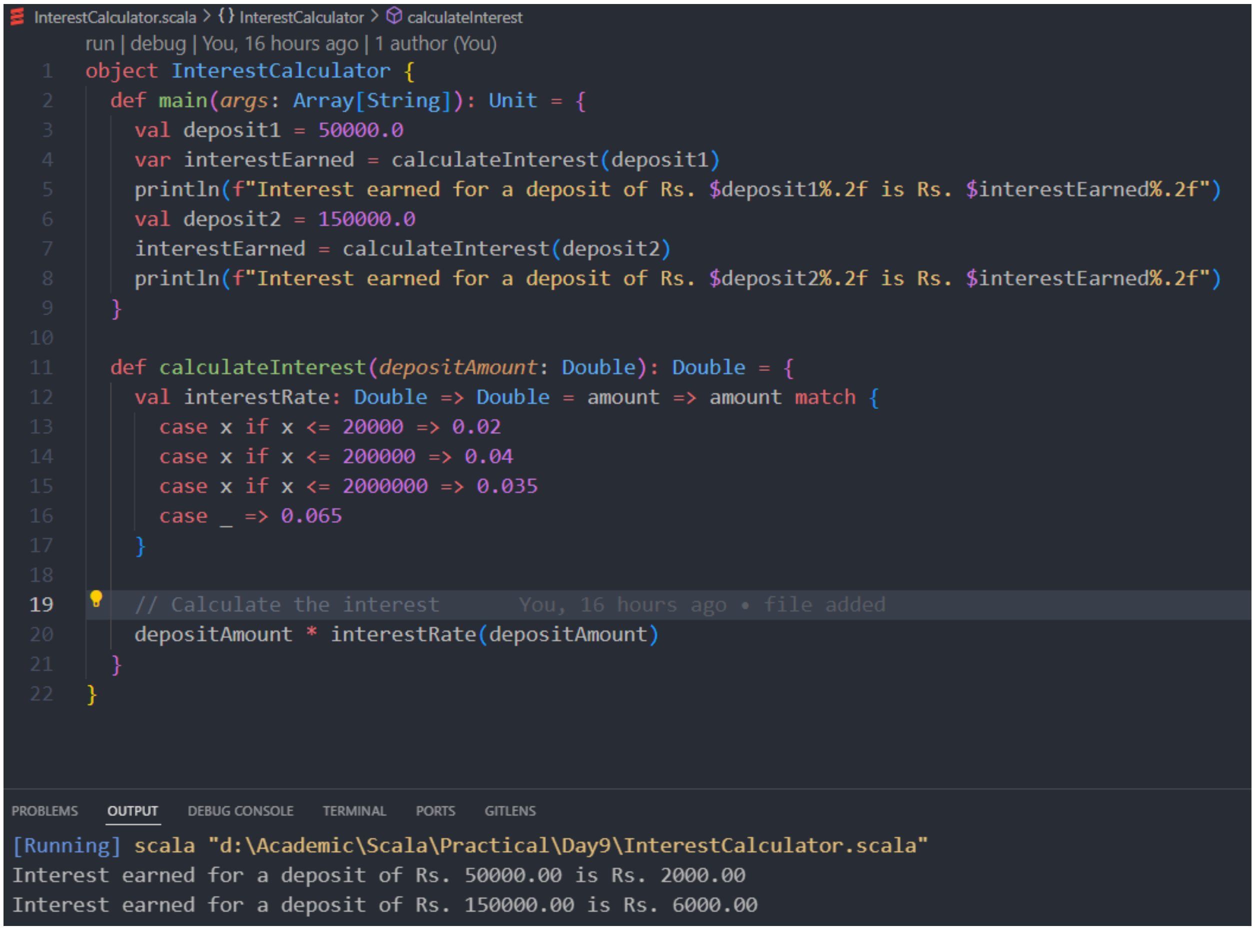
LabSheet 9

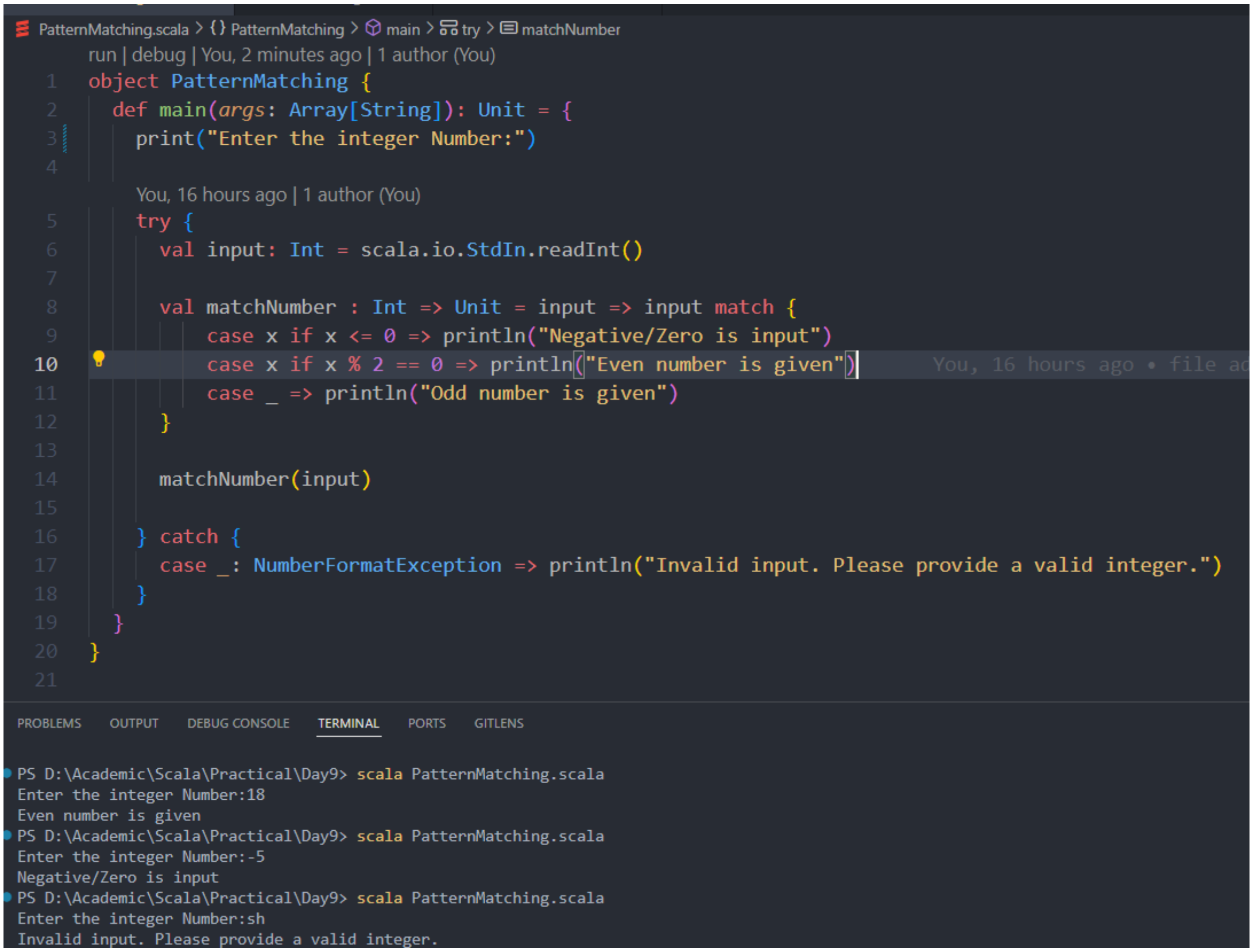
1)


InterestCalculator.scala > {} InterestCalculator > calculateInterest
run | debug | You, 16 hours ago | 1 author (You)
You, 16 hours ago • file added
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
GITLENS

```scala
object InterestCalculator {
  def main(args: Array[String]): Unit = {
    val deposit1 = 50000.0
    var interestEarned = calculateInterest(deposit1)
    println(f"Interest earned for a deposit of Rs. $deposit1%.2f is Rs. $interestEarned%.2f")
    val deposit2 = 150000.0
    interestEarned = calculateInterest(deposit2)
    println(f"Interest earned for a deposit of Rs. $deposit2%.2f is Rs. $interestEarned%.2f")
  }

  def calculateInterest(depositAmount: Double): Double = {
    val interestRate: Double => Double = amount => amount match {
      case x if x <= 20000 => 0.02
      case x if x <= 200000 => 0.04
      case x if x <= 2000000 => 0.035
      case _ => 0.065
    }

    // Calculate the interest
    depositAmount * interestRate(depositAmount)
  }
}
```

```
[Running] scala "d:\Academic\Scala\Practical\Day9\InterestCalculator.scala"
Interest earned for a deposit of Rs. 50000.00 is Rs. 2000.00
Interest earned for a deposit of Rs. 150000.00 is Rs. 6000.00
```

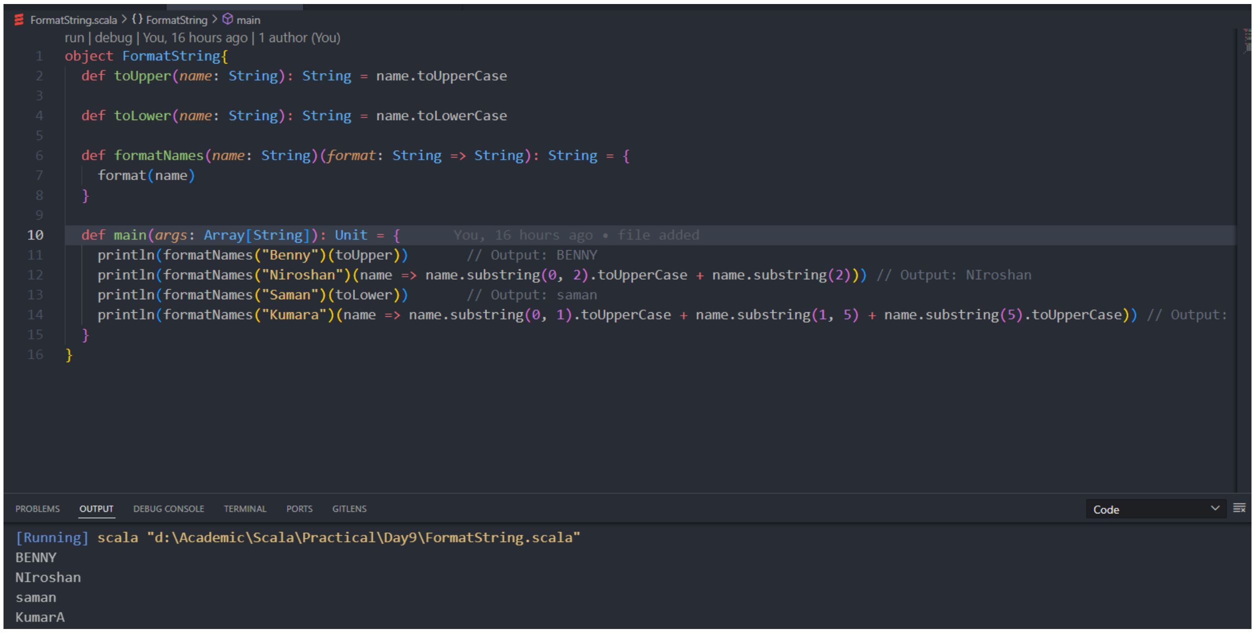
2)


PatternMatching.scala > {} PatternMatching > main > try > matchNumber
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
GITLENS

```
run | debug | You, 2 minutes ago | 1 author (You)
object PatternMatching {
  def main(args: Array[String]): Unit = {
    print("Enter the integer Number:")

    You, 16 hours ago | 1 author (You)
    try {
      val input: Int = scala.io.StdIn.readInt()

      val matchNumber : Int => Unit = input => input match {
        case x if x <= 0 => println("Negative/Zero is input")
        case x if x % 2 == 0 => println("Even number is given")        You, 16 hours ago • file ad
        case _ => println("Odd number is given")
      }

      matchNumber(input)

    } catch {
      case _: NumberFormatException => println("Invalid input. Please provide a valid integer.")
    }
  }
}
```

```
PS D:\Academic\Scala\Practical\Day9> scala PatternMatching.scala
Enter the integer Number:18
Even number is given
PS D:\Academic\Scala\Practical\Day9> scala PatternMatching.scala
Enter the integer Number:-5
Negative/Zero is input
PS D:\Academic\Scala\Practical\Day9> scala PatternMatching.scala
Enter the integer Number:sh
Invalid input. Please provide a valid integer.
```

3)



```
run | debug | You, 16 hours ago | 1 author (You)
object FormatString{
  def toUpper(name: String): String = name.toUpperCase

  def toLower(name: String): String = name.toLowerCase

  def formatNames(name: String)(format: String => String): String = {
    format(name)
  }

  def main(args: Array[String]): Unit = {      You, 16 hours ago • file added
    println(formatNames("Benny")(toUpper))       // Output: BENNY
    println(formatNames("Niroshan")(name => name.substring(0, 2).toUpperCase + name.substring(2))) // Output: NIroshan
    println(formatNames("Saman")(toLower))       // Output: saman
    println(formatNames("Kumara")(name => name.substring(0, 1).toUpperCase + name.substring(1, 5) + name.substring(5).toUpperCase)) // Output:
  }
}
```

```
[Running] scala "d:\Academic\Scala\Practical\Day9\FormatString.scala"
BENNY
NIroshan
saman
KumarA
```