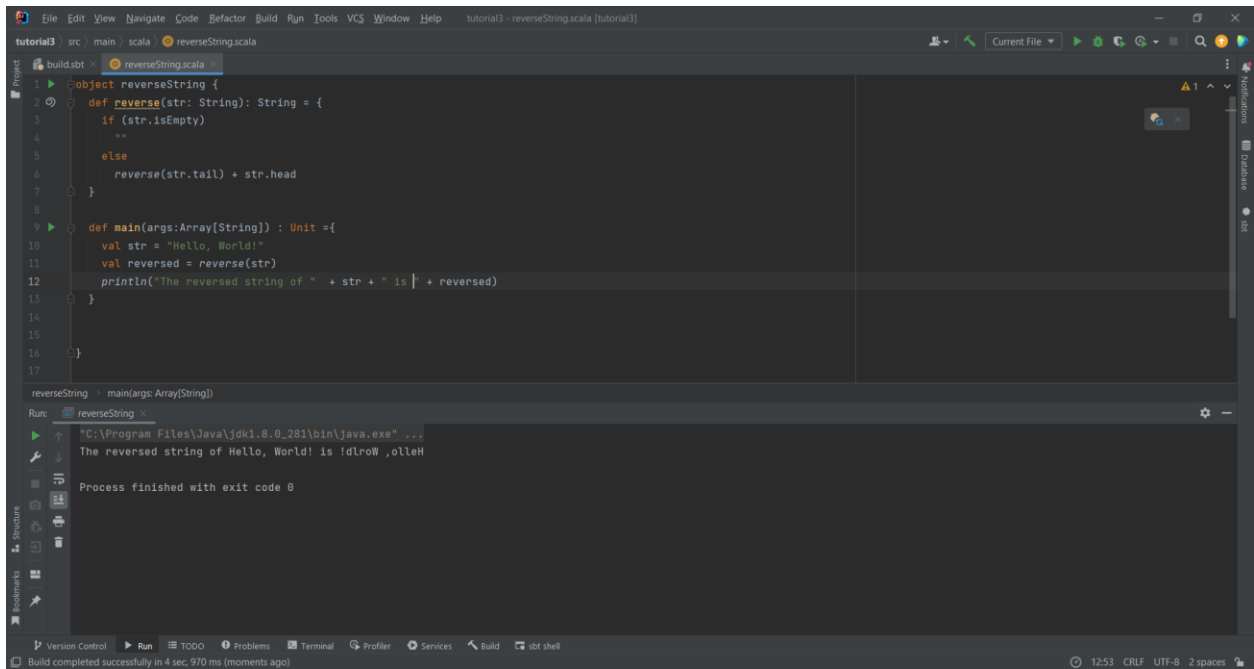


Functional Programming – Tutorial 3

Question 1



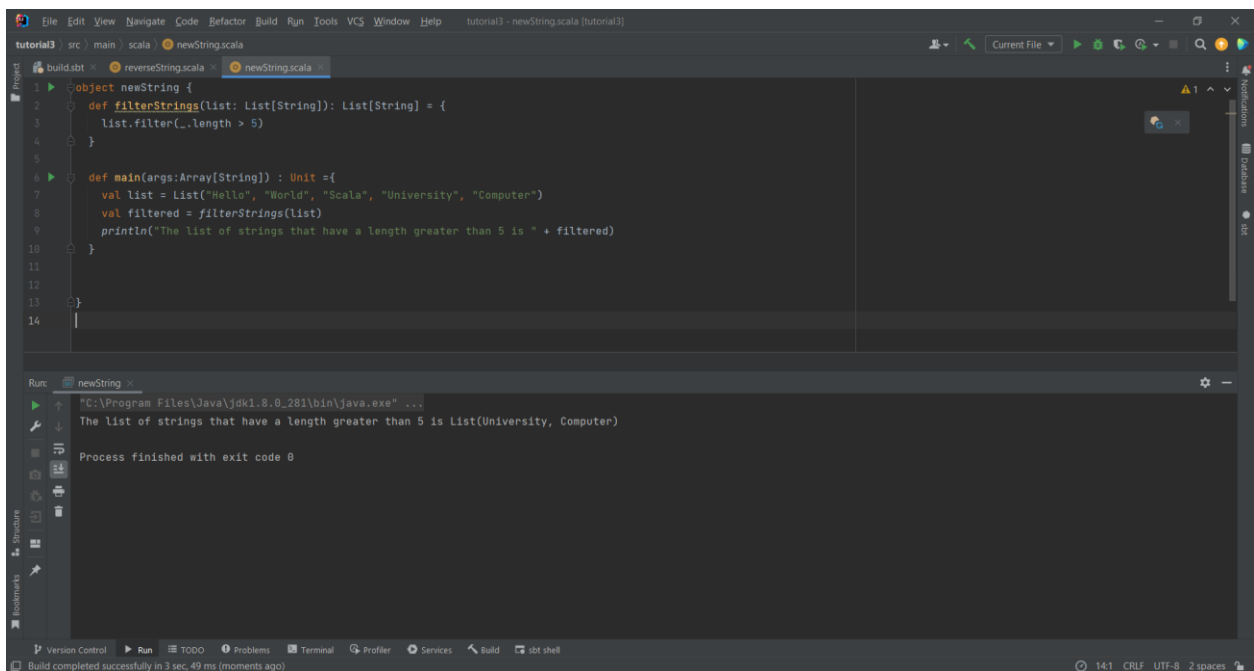
The screenshot shows an IDE with a Scala file named `reverseString.scala`. The code defines an object `reverseString` with a recursive function `reverse` and a `main` method. The `reverse` function takes a `String` and returns its reverse. The `main` method prints the reversed string of "Hello, World!". The output window shows the execution result: "The reversed string of Hello, World! is !dlroW ,olleH".

```
object reverseString {  
  def reverse(str: String): String = {  
    if (str.isEmpty)  
      ""  
    else  
      reverse(str.tail) + str.head  
  }  
  
  def main(args: Array[String]) : Unit = {  
    val str = "Hello, World!"  
    val reversed = reverse(str)  
    println("The reversed string of " + str + " is " + reversed)  
  }  
}
```

Run: reverseString

Process finished with exit code 0

Question 2



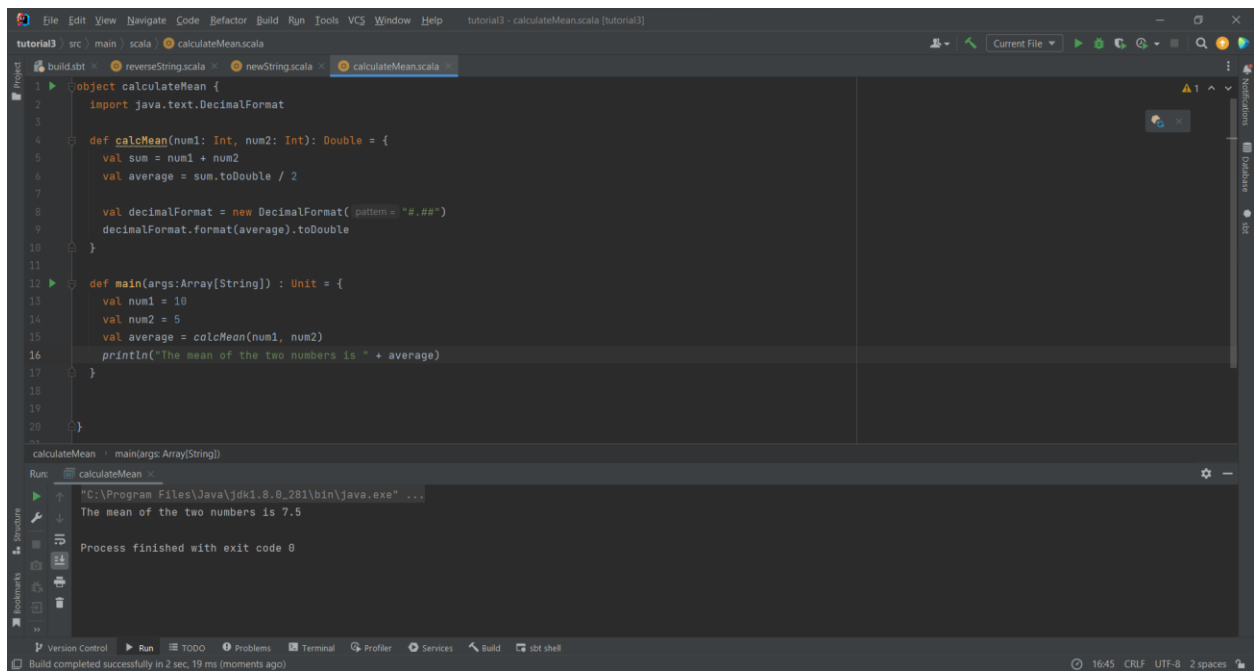
The screenshot shows an IDE with a Scala file named `newString.scala`. The code defines an object `newString` with a function `filterStrings` and a `main` method. The `filterStrings` function takes a `List[String]` and returns a new list containing only the strings with a length greater than 5. The `main` method prints the filtered list. The output window shows the execution result: "The List of strings that have a length greater than 5 is List(University, Computer)".

```
object newString {  
  def filterStrings(list: List[String]): List[String] = {  
    list.filter(_.length > 5)  
  }  
  
  def main(args: Array[String]) : Unit = {  
    val list = List("Hello", "World", "Scala", "University", "Computer")  
    val filtered = filterStrings(list)  
    println("The List of strings that have a length greater than 5 is " + filtered)  
  }  
}
```

Run: newString

Process finished with exit code 0

Question 3



The screenshot shows an IDE window titled 'tutorial3 - calculateMean.scala [tutorial3]'. The code defines an object `calculateMean` with a method `calcMean` that takes two integers and returns a double. It also includes a `main` method that calls `calcMean` with values 10 and 5, and prints the result. The output window shows the command `calculateMean - main(args: Array[String])` and the output `The mean of the two numbers is 7.5`. The status bar at the bottom indicates 'Build completed successfully in 2 sec, 19 ms (moments ago)'.

```
1 object calculateMean {
2   import java.text.DecimalFormat
3
4   def calcMean(num1: Int, num2: Int): Double = {
5     val sum = num1 + num2
6     val average = sum.toDouble / 2
7
8     val decimalFormat = new DecimalFormat(pattern = "#.##")
9     decimalFormat.format(average).toDouble
10  }
11
12  def main(args: Array[String]) : Unit = {
13    val num1 = 10
14    val num2 = 5
15    val average = calcMean(num1, num2)
16    println("The mean of the two numbers is " + average)
17  }
18
19 }
20
```

calculateMean - main(args: Array[String])

Run: calculateMean

"C:\Program Files\Java\jdk1.8.0_281\bin\java.exe" ...
The mean of the two numbers is 7.5

Process finished with exit code 0

Build completed successfully in 2 sec, 19 ms (moments ago)