

## 5. Problem Statement

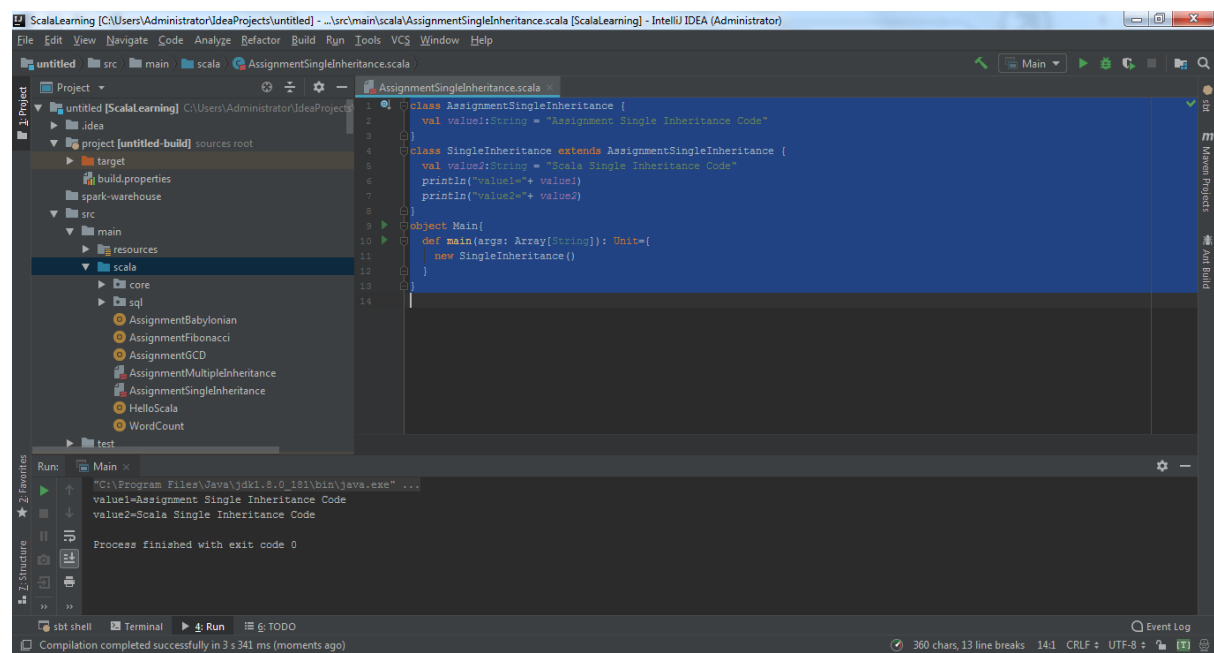
### Task 1

Write a simple program to show inheritance in scala.

Scala Code:-

```
class AssignmentSingleInheritance {  
    val value1:String = "Assignment Single Inheritance Code"  
}  
class SingleInheritance extends AssignmentSingleInheritance {  
    val value2:String = "Scala Single Inheritance Code"  
    println("value1="+ value1)  
    println("value2="+ value2)  
}  
object Main{  
    def main(args: Array[String]): Unit={  
        new SingleInheritance()  
    }  
}
```

Scala Output:-



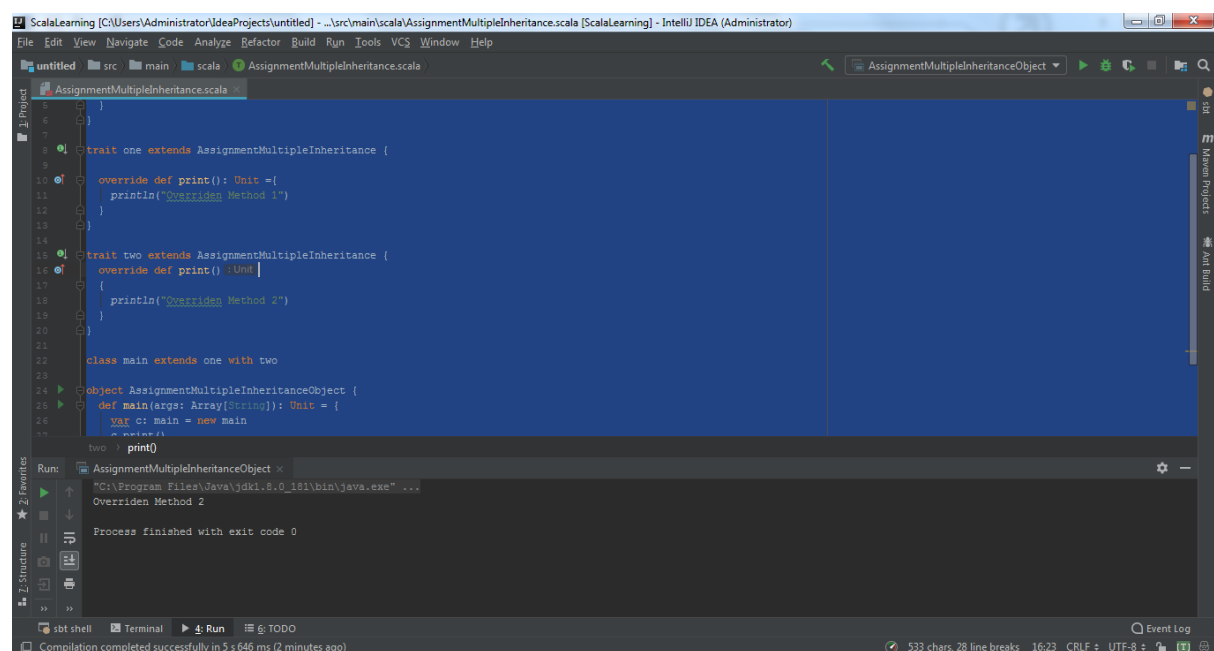
## Task 2

Write a simple program to show multiple inheritance in scala

Scala Code:-

```
trait AssignmentMultipleInheritance {  
  
    def print(): Unit = {  
        println("Acadgild Multiple Inheritance Assignment")  
    }  
}  
  
trait one extends AssignmentMultipleInheritance {  
  
    override def print(): Unit = {  
        println("Overriden Method 1")  
    }  
}  
  
trait two extends AssignmentMultipleInheritance {  
    override def print()  
    {  
        println("Overriden Method 2")  
    }  
}  
  
class main extends one with two  
  
object AssignmentMultipleInheritanceObject {  
    def main(args: Array[String]): Unit = {  
        var c: main = new main  
        c.print()  
    }  
}
```

Scala Output:-



The screenshot displays the IntelliJ IDEA IDE interface. The main editor window shows the Scala code from the previous block, with line numbers 1 through 27. The code defines a trait `AssignmentMultipleInheritance`, two traits `one` and `two` that extend it, and a class `main` that extends both `one` and `two`. An object `AssignmentMultipleInheritanceObject` contains a `main` method that creates an instance of `main` and calls its `print` method. The `Run` tab at the bottom shows the execution output: `Overriden Method 2`. The status bar at the bottom indicates that the compilation was successful in 5 s 646 ms (2 minutes ago).

### Task 3

Write a partial function to add three numbers in which one number is constant and two

numbers can be passed as inputs and define another method which can take the partial

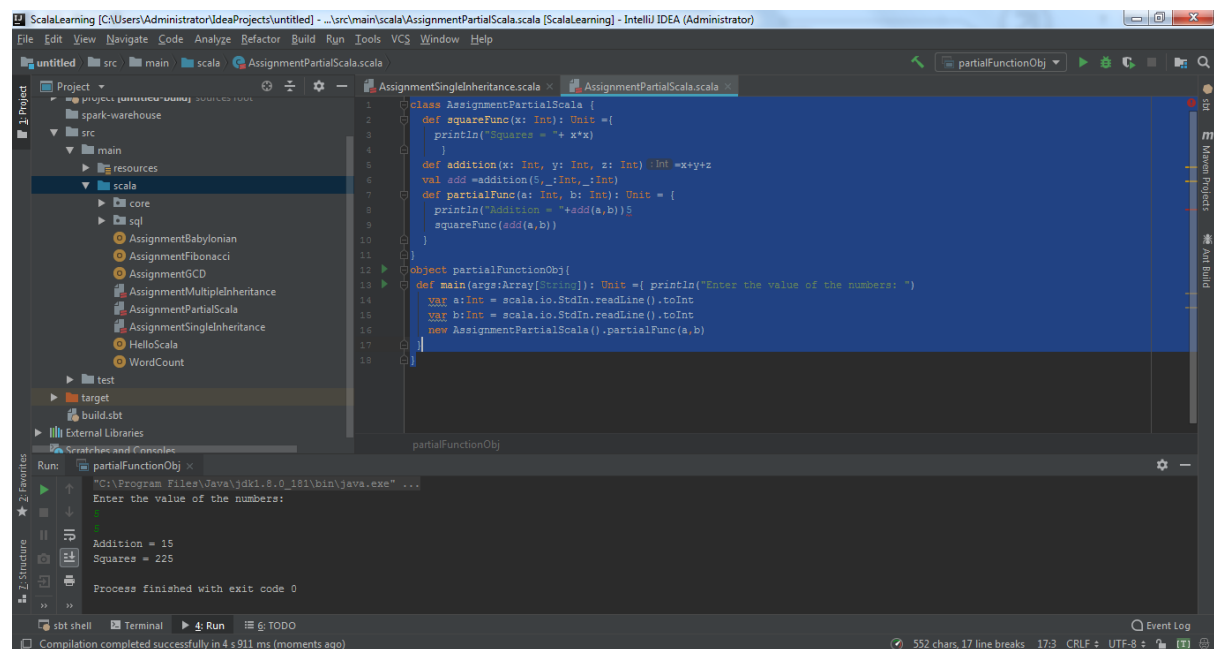
function as input and squares the result.

Scala Code:-

```
class AssignmentPartialScala {
  def squareFunc(x: Int): Unit = {
    println("Squares = " + x*x)
  }
  def addition(x: Int, y: Int, z: Int) = x+y+z
  val add = addition(5, _: Int, _: Int)
  def partialFunc(a: Int, b: Int): Unit = {
    println("Addition = " + add(a,b))
    squareFunc(add(a,b))
  }
}

object partialFunctionObj {
  def main(args: Array[String]): Unit = {
    println("Enter the value of the numbers: ")
    var a: Int = scala.io.StdIn.readLine().toInt
    var b: Int = scala.io.StdIn.readLine().toInt
    new AssignmentPartialScala().partialFunc(a,b)
  }
}
```

Scala Output:-



## Task 4

Write a program to print the prices of 4 courses of Acadgild:

Android App Development -14,999 INR

Data Science - 49,999 INR

Big Data Hadoop & Spark Developer – 24,999 INR

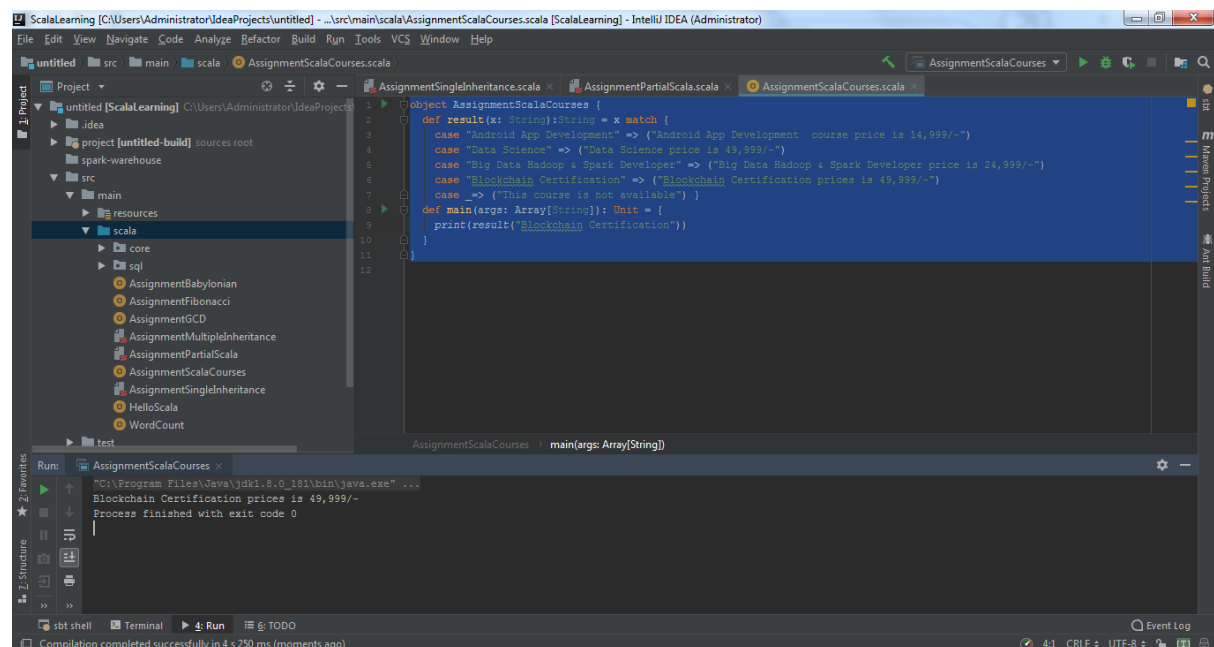
Blockchain Certification – 49,999 INR

using match and add a default condition if the user enters any other course.

Scala Code:-

```
object AssignmentScalaCourses {  
  def result(x: String):String = x match {  
    case "Android App Development" => ("Android App Development course price is 14,999/-")  
    case "Data Science" => ("Data Science price is 49,999/-")  
    case "Big Data Hadoop & Spark Developer" => ("Big Data Hadoop & Spark Developer price is 24,999/-")  
    case "Blockchain Certification" => ("Blockchain Certification prices is 49,999/-")  
    case _ => ("This course is not available") }  
  def main(args: Array[String]): Unit = {  
    print(result("Blockchain Certification"))  
  }  
}
```

Scala Output:-



The screenshot shows the IntelliJ IDEA interface. The main editor displays the Scala code from the previous block. The left sidebar shows the project structure with the 'scala' directory selected. The bottom panel shows the 'Run' output, which displays the output of the program: 'Blockchain Certification prices is 49,999/-'. The status bar at the bottom indicates 'Compilation completed successfully in 4 s 250 ms (moments ago)'.