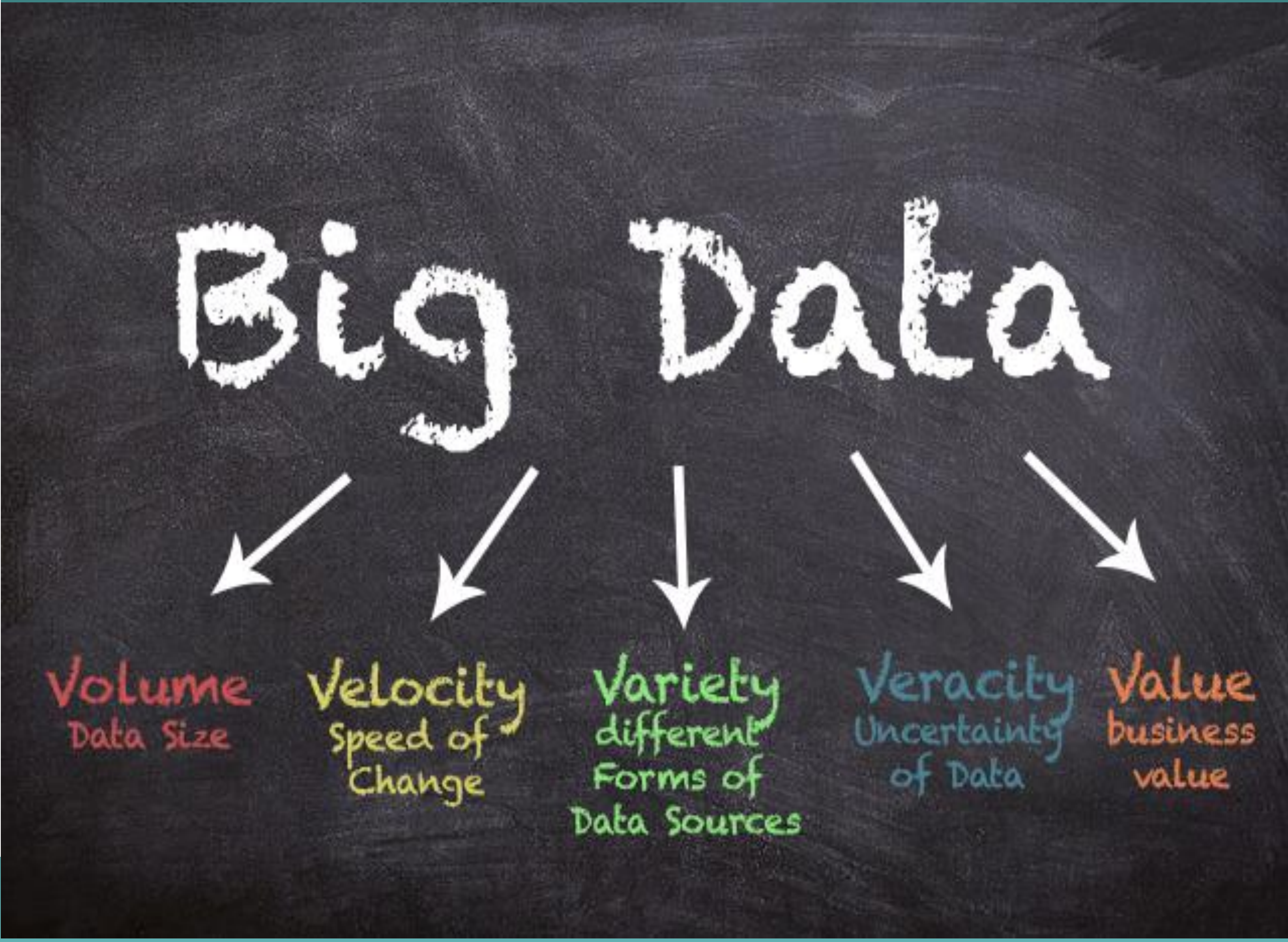
08/22/2018 | By: Ajit K Prasad



Big Data Engineering with Hadoop & Spark

Assignment on Scala Basics

Scala

APACHE Spark

Acadgild

Session 17: Assignment 17.1

This assignment is aimed at consolidating the concepts that was learnt during the Scala Basics session of the course.

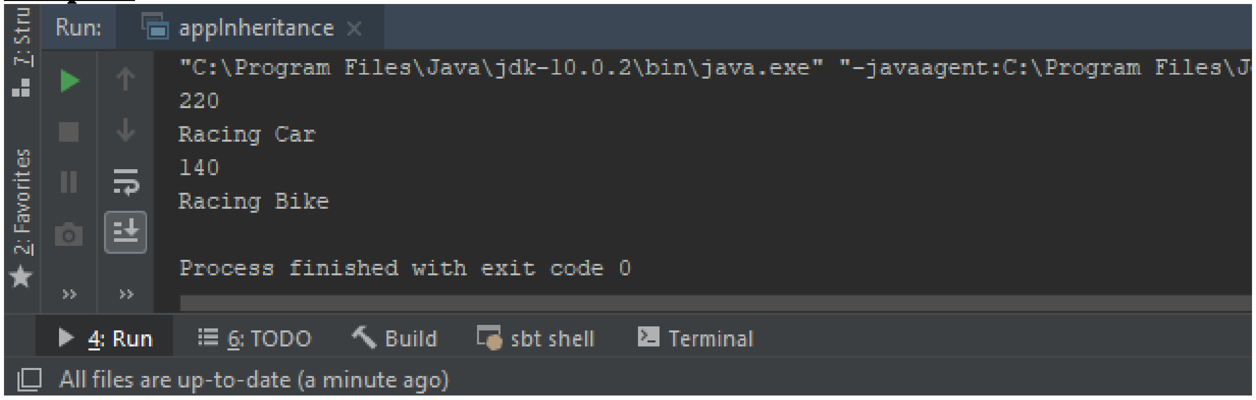
Task 1:

Write a simple program to show inheritance in Scala.

Solution:

```
class Vehicle (speed : Int){
  val mph :Int = speed
  def race() = println("Racing")
}

class Car (speed : Int) extends Vehicle(speed) {
  override val mph: Int= speed
  override def race() = println("Racing Car")

}
class Bike(speed : Int) extends Vehicle(speed) {
  override val mph: Int = speed
  override def race() = println("Racing Bike")
}
object appInheritance extends App {
  val vehicle1 = new Car(220)
  println(vehicle1.mph )
  vehicle1.race()

  val vehicle2 = new Bike(140)
  println(vehicle2.mph )
  vehicle2.race()
}
```

Output:



```
"C:\Program Files\Java\jdk-10.0.2\bin\java.exe" "-javaagent:C:\Program Files\J
220
Racing Car
140
Racing Bike

Process finished with exit code 0
```

All files are up-to-date (a minute ago)

Task 2:

Write a simple program to show multiple inheritance in Scala.

Solution:

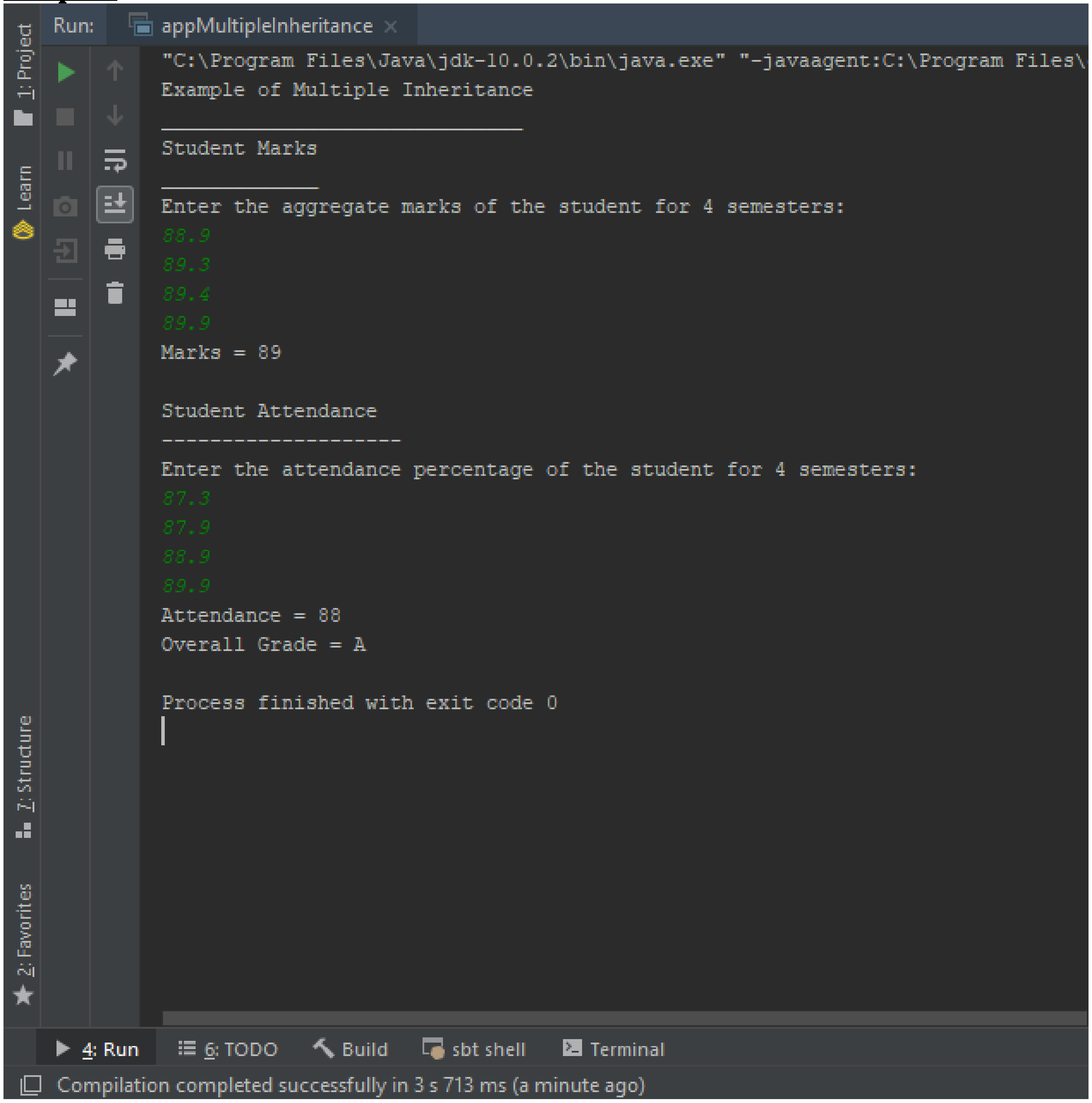
To show multiple inheritance in Scala, we use a concept called Trait. Scala Traits consists of method and field definitions that can be reused by mixing classes. The class can mix any number of traits.

Code:

```
trait Aggregate {
  def sum(x:Array[Double]): Double = {
    var sum:Double = 0
    for (i <- 0 to x.length-1) sum = sum + x(i)
    sum
  }
  def sum(w: Double, x: Double, y: Double, z: Double): Double = w + x + y +z
}

trait Average{
  def avg(sum: Double, length: Int): Double = sum/length
  def analyzeAvg(avg: Int): String = {
    var grade: String = ""
    if(avg > 85) grade = "A"
    else if(avg < 85 && avg > 65) grade = "B"
    else if(avg < 65 && avg > 40) grade = "C"
    else grade = "F"
    grade
  }
}

class Student extends Aggregate with Average {
  var marks:Array[Double] = new Array[Double] (4)
  var overallGrade = 0
  def StudentMarks(): Unit = {
    println("Student Marks")
    println("_____________")
    println("Enter the aggregate marks of the student for 4 semesters: ")
    for(y <- 0 to marks.length-1)
      marks(y) = scala.io.StdIn.readLine().toDouble
    var finalSum = sum(marks)
    var finalAvg = avg(finalSum, marks.length).toInt
    println(s"Marks = " + finalAvg)
    overallGrade = overallGrade + finalAvg
  }
  def StudentAttendance(): Unit = {
    println("\nStudent Attendance")
    println("--------------------")
    println("Enter the attendance percentage of the student for 4 semesters: ")
    var sem1a = scala.io.StdIn.readLine().toDouble
    var sem2a = scala.io.StdIn.readLine().toDouble
    var sem3a = scala.io.StdIn.readLine().toDouble
    var sem4a = scala.io.StdIn.readLine().toDouble
    var finalSum = sum(sem1a, sem2a, sem3a, sem4a)
    var finalAvg = avg(finalSum,4).toInt
    println(s"Attendance = " + finalAvg)
    overallGrade = overallGrade + finalAvg
  }
  def Grade(): Unit = println(s"Overall Grade = " + analyzeAvg(overallGrade/2))
}
```

```
object appMultipleInheritance{
  def main(args: Array[String]) {
    println("Example of Multiple Inheritance")
    println("______________________________")
    var sDetails:Student = new Student()
    sDetails.StudentMarks()
    sDetails.StudentAttendance()
    sDetails.Grade()
  }
}
```

Output:



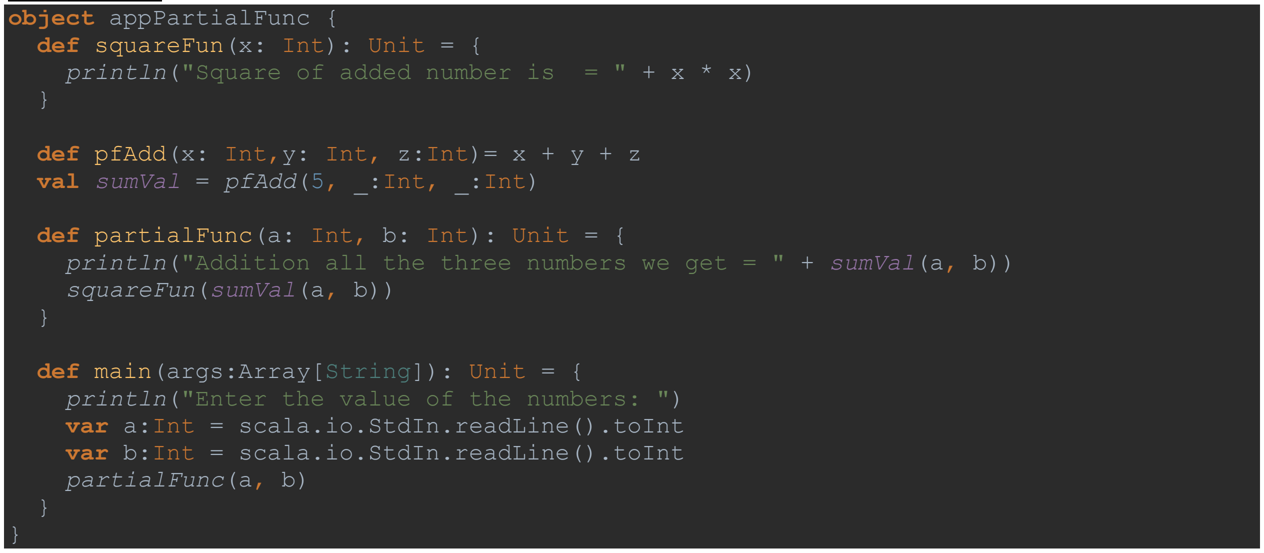
```
"C:\Program Files\Java\jdk-10.0.2\bin\java.exe" "-javaagent:C:\Program Files\
Example of Multiple Inheritance

______________________________
Student Marks

_____________
Enter the aggregate marks of the student for 4 semesters:
88.9
89.3
89.4
89.9
Marks = 89

Student Attendance
--------------------
Enter the attendance percentage of the student for 4 semesters:
87.3
87.9
88.9
89.9
Attendance = 88
Overall Grade = A

Process finished with exit code 0
```

Compilation completed successfully in 3 s 713 ms (a minute ago)

Task 3:

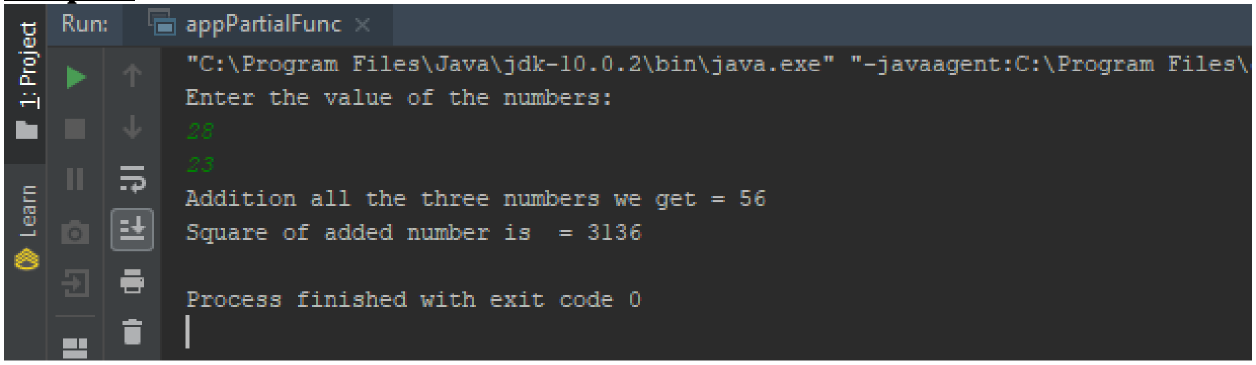
Write a partial function to add three numbers in which one number is constant and two numbers can be passed as inputs and define another method which can take the partial function as input and squares the result.

Solution:



```
object appPartialFunc {
  def squareFun(x: Int): Unit = {
    println("Square of added number is  = " + x * x)
  }

  def pfAdd(x: Int,y: Int, z:Int)= x + y + z
  val sumVal = pfAdd(5, _:Int, _:Int)

  def partialFunc(a: Int, b: Int): Unit = {
    println("Addition all the three numbers we get = " + sumVal(a, b))
    squareFun(sumVal(a, b))
  }

  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
    partialFunc(a, b)
  }
}
```

Output:



```
"C:\Program Files\Java\jdk-10.0.2\bin\java.exe" "-javaagent:C:\Program Files\
Enter the value of the numbers:
28
23
Addition all the three numbers we get = 56
Square of added number is  = 3136

Process finished with exit code 0
```

Task 4:

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

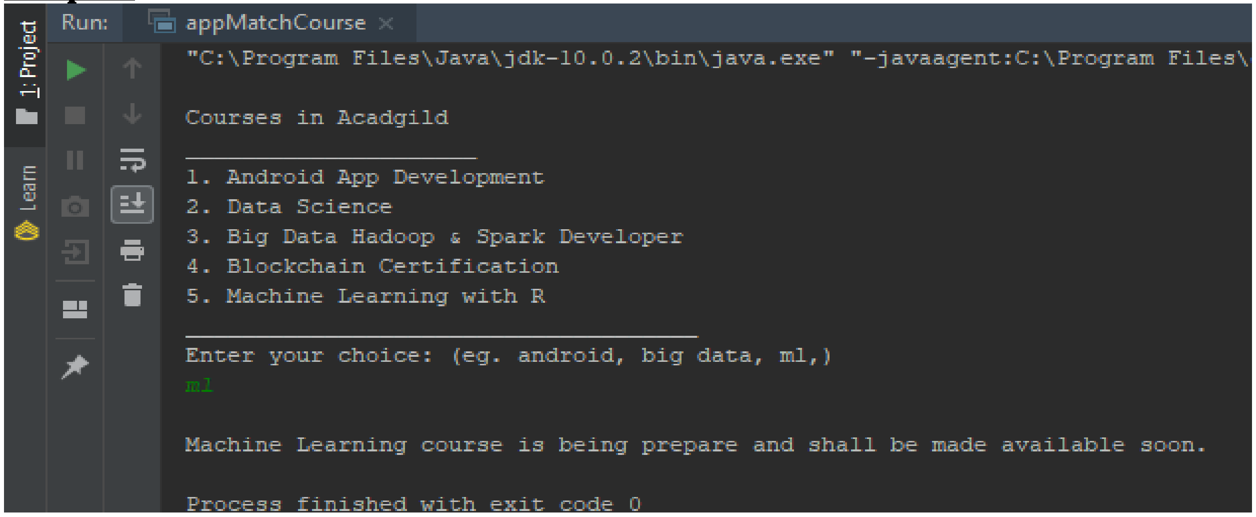
– Android App Development -14,999 INR
– Data Science - 49,999 INR
– Big Data Hadoop & Spark Developer – 24,999 INR
– Blockchain Certification – 49,999 INR

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

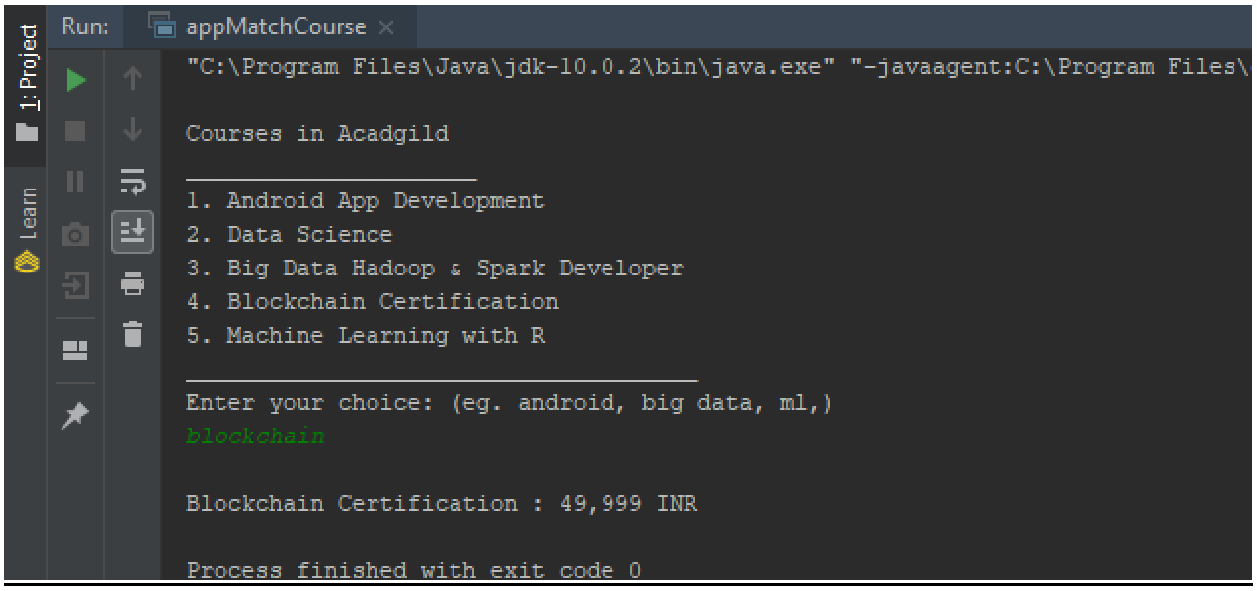
Solution:

```scala
object appMatchCourse {
  def courseMatch(course:String): String = course match  {
      case "android" => "Android App Development : 14,999 INR"
      case "data science" => "Data Science : 49,999 INR"
      case "big data" => "Big Data & Spark course is : 24,999 INR"
      case "blockchain" => "Blockchain Certification : 49,999 INR"
      case _ => "Machine Learning course is being prepare and shall be made available
soon."
    }
  def main(args:Array[String]): Unit = {
    println("\nCourses in Acadgild")
    println("_____________________")
    println("1. Android App Development")
    println("2. Data Science")
    println("3. Big Data Hadoop & Spark Developer")
    println("4. Blockchain Certification")
    println("5. Machine Learning with R")
    println("____________________________________")
    println("Enter your choice: (eg. android, big data, ml,) " )
    var choice = scala.io.StdIn.readLine().toString.toLowerCase
    println("\n" + courseMatch(choice))
  }
}
```

Output:



```
"C:\Program Files\Java\jdk-10.0.2\bin\java.exe" "-javaagent:C:\Program Files\

Courses in Acadgild

____________________
1. Android App Development
2. Data Science
3. Big Data Hadoop & Spark Developer
4. Blockchain Certification
5. Machine Learning with R

___________________________________
Enter your choice: (eg. android, big data, ml,)
ml

Machine Learning course is being prepare and shall be made available soon.

Process finished with exit code 0
```



```
"C:\Program Files\Java\jdk-10.0.2\bin\java.exe" "-javaagent:C:\Program Files\

Courses in Acadgild
______________________
1. Android App Development
2. Data Science
3. Big Data Hadoop & Spark Developer
4. Blockchain Certification
5. Machine Learning with R
______________________________________
Enter your choice: (eg. android, big data, ml,)
blockchain

Blockchain Certification : 49,999 INR

Process finished with exit code 0
```

Note:
Scala code files for each application has been provided separately along with this assignment report.