ASSIGNMENT 15.2

Problem Statement:

1. 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.
2. Write a program to print the prices of 4 courses of Acadgild: Android-12999, Big Data Development-17999, Full Stack Development-17999, Spark-19999 using match and add a default condition if the user enters any other course.

Solution:

1. Here is the Scala code for first problem:

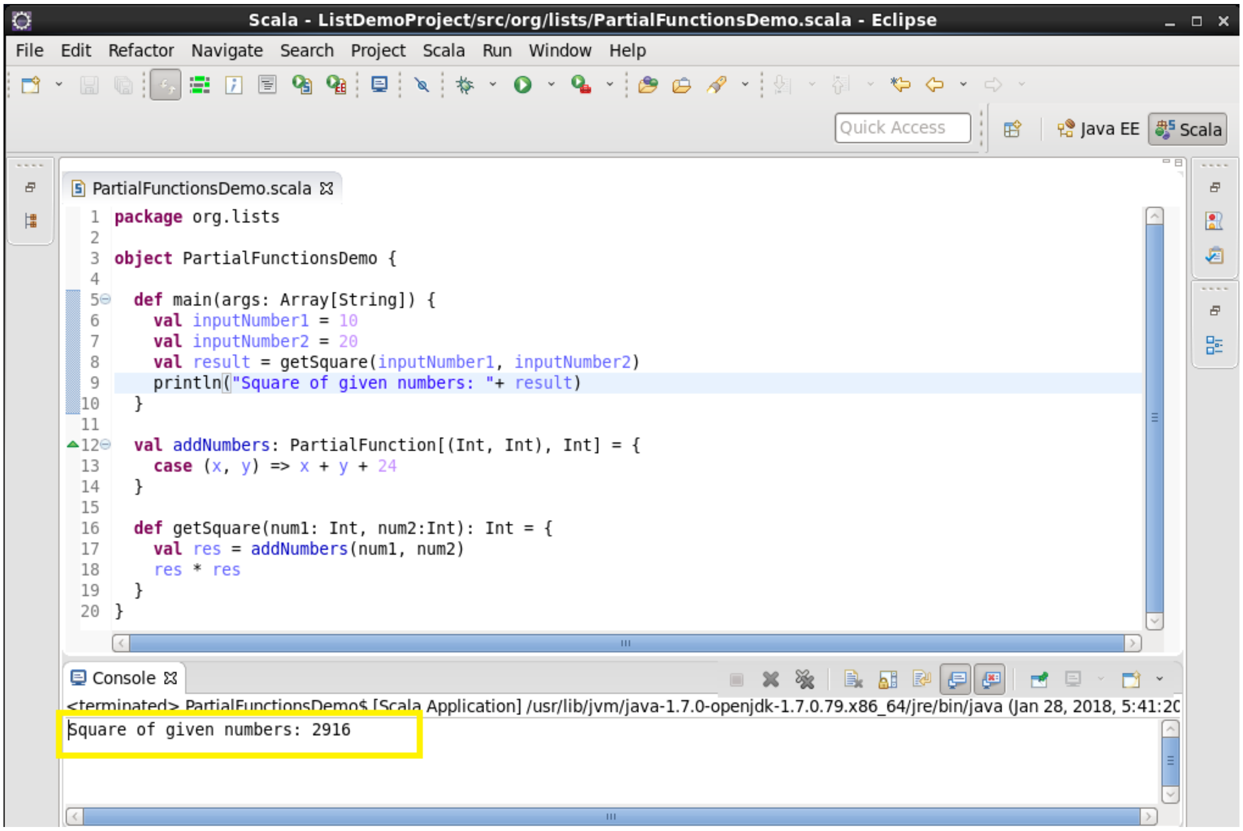
PartialFunctionsDemo.scala

```
object PartialFunctionsDemo {
    def main(args: Array[String]) {
        val inputNumber1 = 10                                    // input values
        val inputNumber2 = 20
        val result = getSquare(inputNumber1, inputNumber2)   // call function to get required result
        println("Square of given numbers: " + result)            // print result
    }
    val addNumbers: PartialFunction[(Int, Int), Int] = {          // partial function to compute sum
        case (x, y) => x + y + 24                              // add two input values with a constant
    }
    def getSquare(num1: Int, num2:Int): Int = {
        val res = addNumbers(num1, num2)                       // call partial function to get the sum
        res * res                                              // return square of three numbers
    }
}
```

Output:

Square of given numbers: 2916

P.S.: Input numbers – 10, 20 and 24


Scala - ListDemoProject/src/org/lists/PartialFunctionsDemo.scala - Eclipse
File Edit Refactor Navigate Search Project Scala Run Window Help
Quick Access
Java EE
Scala
PartialFunctionsDemo.scala
Console

```
package org.lists

object PartialFunctionsDemo {

  def main(args: Array[String]) {
    val inputNumber1 = 10
    val inputNumber2 = 20
    val result = getSquare(inputNumber1, inputNumber2)
    println("Square of given numbers: "+ result)
  }

  val addNumbers: PartialFunction[(Int, Int), Int] = {
    case (x, y) => x + y + 24
  }

  def getSquare(num1: Int, num2:Int): Int = {
    val res = addNumbers(num1, num2)
    res * res
  }
}
```

```
<terminated> PartialFunctionsDemo$ [Scala Application] /usr/lib/jvm/java-1.7.0-openjdk-1.7.0.79.x86_64/jre/bin/java (Jan 28, 2018, 5:41:2C
Square of given numbers: 2916
```

2. Here is the Scala code for first problem:

MatchCaseDemo.scala

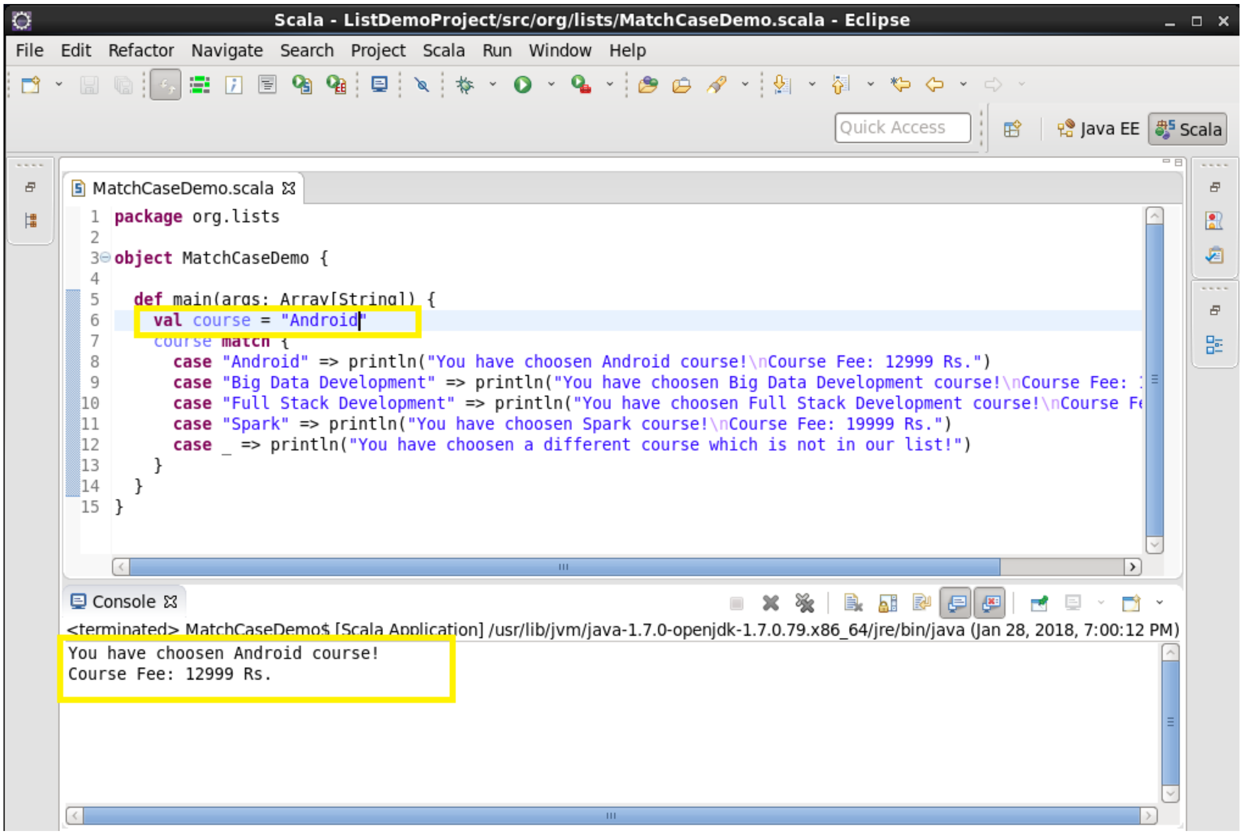
```
object MatchCaseDemo {
  def main(args: Array[String]) {
    val course = " Android"                         // input value for ‘course’ from user
    course match {                                  // match expression
      case "Android" => println("You have choosen Android course!\nCourse Fee: 12999 Rs.")
      case "Big Data Development" => println("You have choosen Big Data Development course!\nCourse Fee: 17999 Rs.")
    case "Full Stack Development" => println("You have choosen Full Stack Development course!\nCourse Fee: 17999 Rs.")

    case "Spark" => println("You have choosen Spark course!\nCourse Fee: 19999 Rs.")

    case _ => println("You have choosen a different course which is not in our list!") // default case

   }

  }

}
```

Output:

You have choosen Android course!

Course Fee: 12999 Rs.

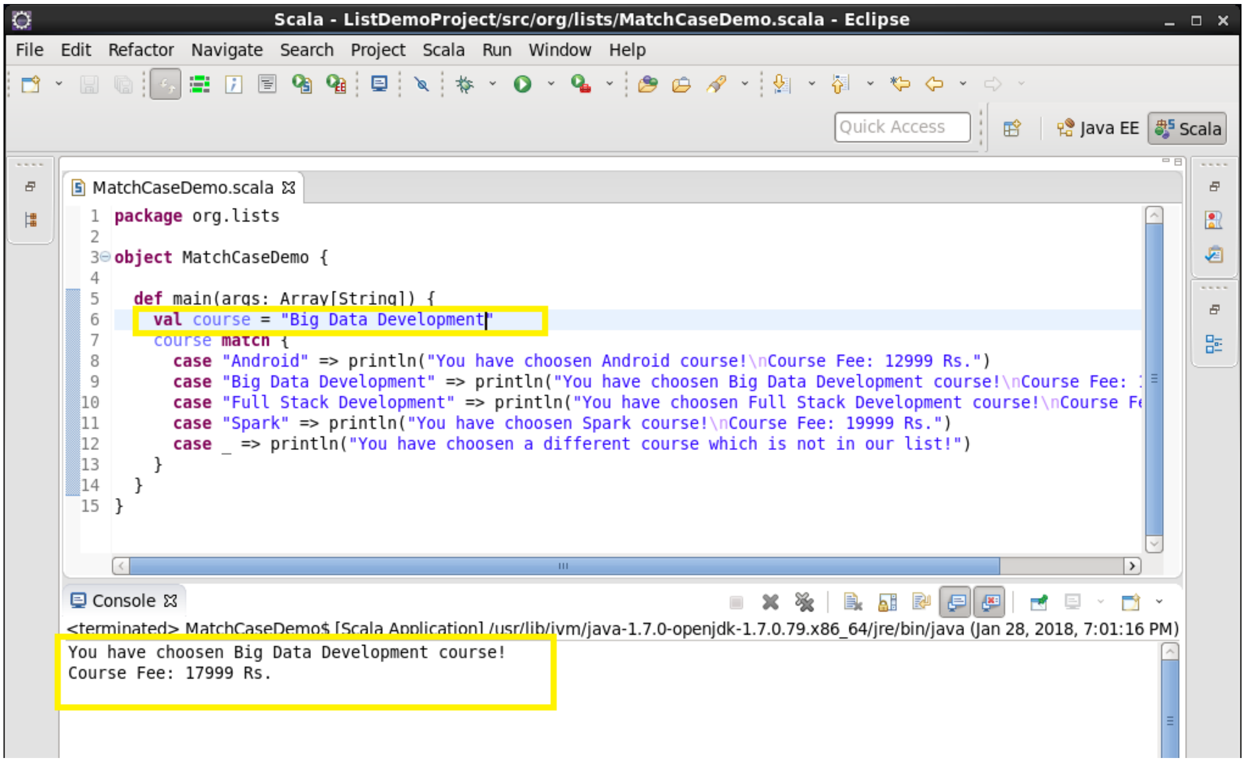
Case 1 output screenshot:



```
package org.lists

object MatchCaseDemo {

  def main(args: Array[String]) {
    val course = "Android"
    course match {
      case "Android" => println("You have choosen Android course!\nCourse Fee: 12999 Rs.")
      case "Big Data Development" => println("You have choosen Big Data Development course!\nCourse Fee: 
      case "Full Stack Development" => println("You have choosen Full Stack Development course!\nCourse Fe
      case "Spark" => println("You have choosen Spark course!\nCourse Fee: 19999 Rs.")
      case _ => println("You have choosen a different course which is not in our list!")
    }
  }
}
```

```
<terminated> MatchCaseDemo$ [Scala Application] /usr/lib/jvm/java-1.7.0-openjdk-1.7.0.79.x86_64/jre/bin/java (Jan 28, 2018, 7:00:12 PM)
You have choosen Android course!
Course Fee: 12999 Rs.
```

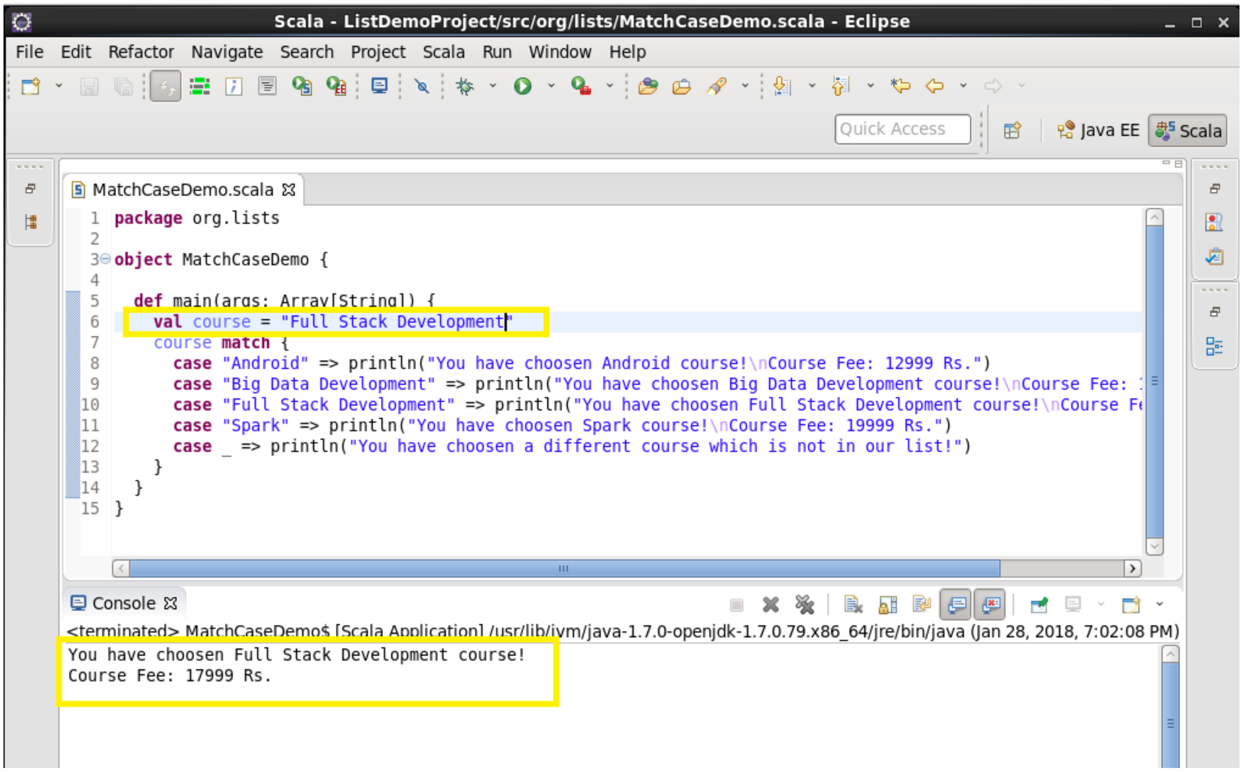
Case 2 output screenshot:



```
package org.lists

object MatchCaseDemo {

  def main(args: Array[String]) {
    val course = "Big Data Development"
    course match {
      case "Android" => println("You have choosen Android course!\nCourse Fee: 12999 Rs.")
      case "Big Data Development" => println("You have choosen Big Data Development course!\nCourse Fee: 
      case "Full Stack Development" => println("You have choosen Full Stack Development course!\nCourse Fe
      case "Spark" => println("You have choosen Spark course!\nCourse Fee: 19999 Rs.")
      case _ => println("You have choosen a different course which is not in our list!")
    }
  }
}
```

```
<terminated> MatchCaseDemo$ [Scala Application] /usr/lib/jvm/java-1.7.0-openjdk-1.7.0.79.x86_64/jre/bin/java (Jan 28, 2018, 7:01:16 PM)
You have choosen Big Data Development course!
Course Fee: 17999 Rs.
```

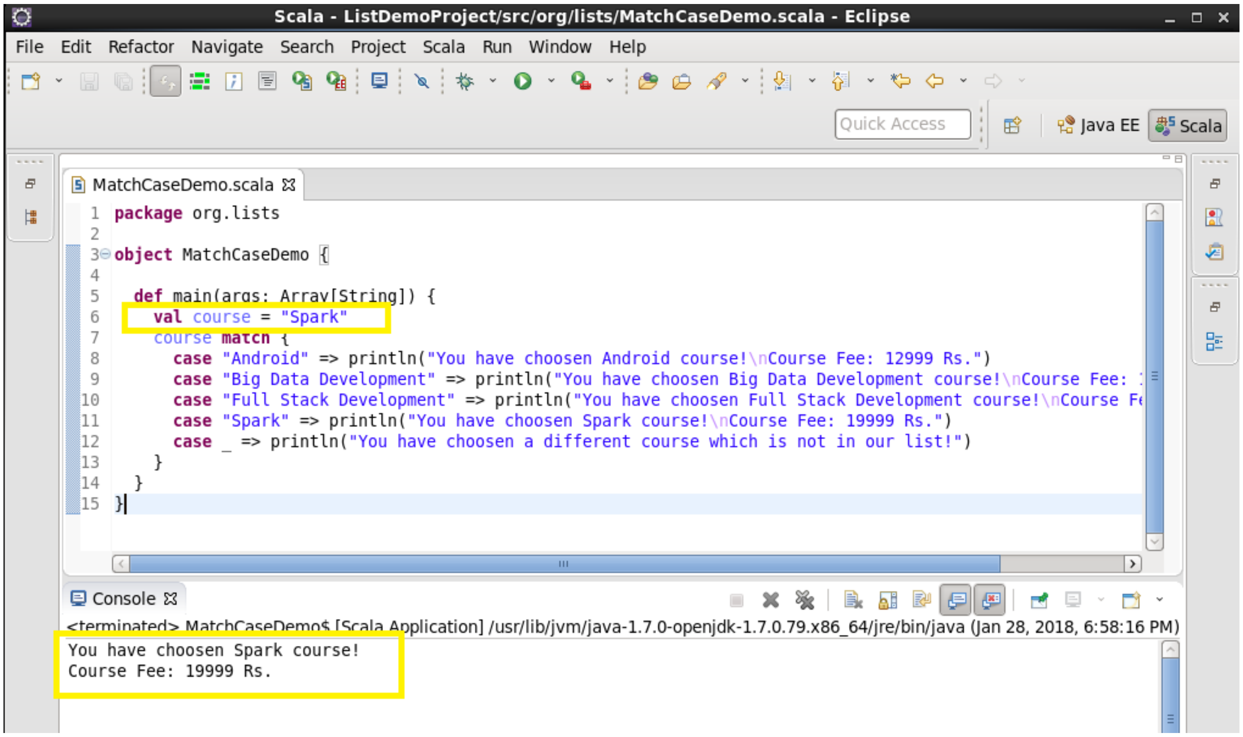
Case 3 output screenshot:



```
package org.lists

object MatchCaseDemo {

  def main(args: Array[String]) {
    val course = "Full Stack Development"
    course match {
      case "Android" => println("You have choosen Android course!\nCourse Fee: 12999 Rs.")
      case "Big Data Development" => println("You have choosen Big Data Development course!\nCourse Fee: 
      case "Full Stack Development" => println("You have choosen Full Stack Development course!\nCourse Fe
      case "Spark" => println("You have choosen Spark course!\nCourse Fee: 19999 Rs.")
      case _ => println("You have choosen a different course which is not in our list!")
    }
  }
}
```

```
<terminated> MatchCaseDemo$ [Scala Application] /usr/lib/jvm/java-1.7.0-openjdk-1.7.0.79.x86_64/jre/bin/java (Jan 28, 2018, 7:02:08 PM)
You have choosen Full Stack Development course!
Course Fee: 17999 Rs.
```

Case 4 output screenshot:

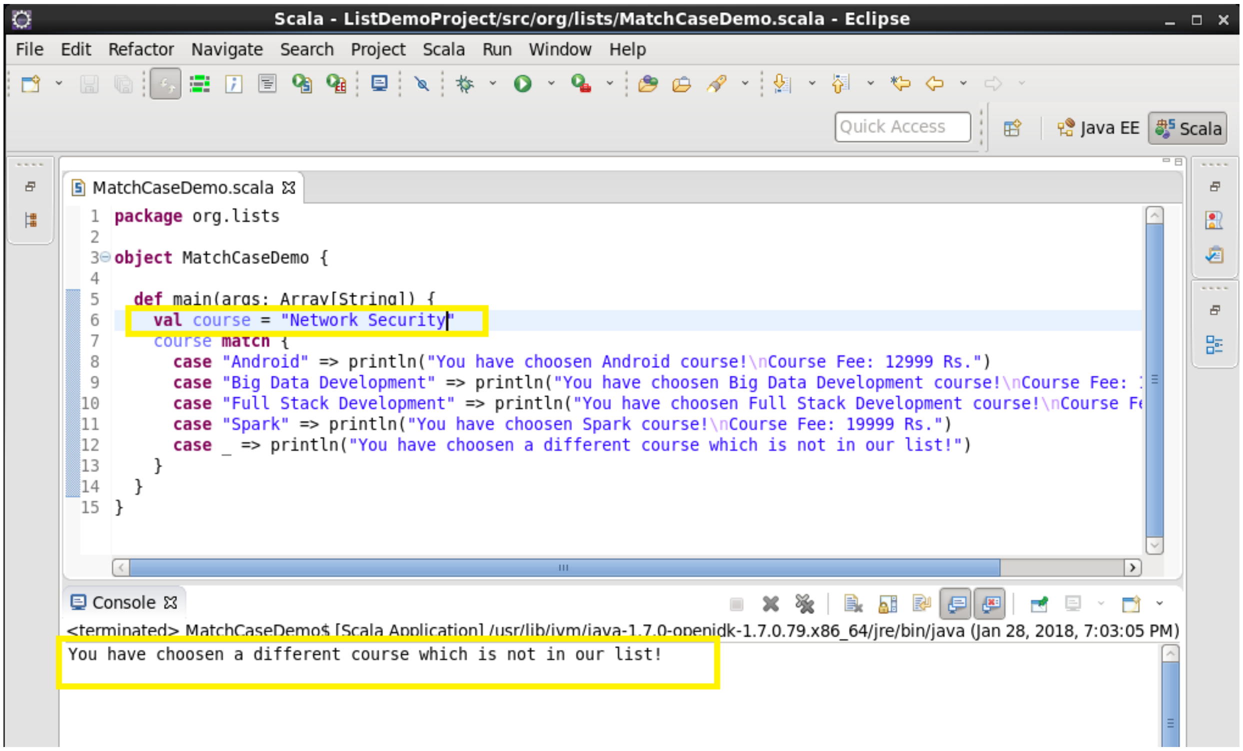


Scala - ListDemoProject/src/org/lists/MatchCaseDemo.scala - Eclipse
File Edit Refactor Navigate Search Project Scala Run Window Help
Quick Access
Java EE
Scala
MatchCaseDemo.scala
Console

```scala
package org.lists

object MatchCaseDemo {

  def main(args: Array[String]) {
    val course = "Spark"
    course match {
      case "Android" => println("You have choosen Android course!\nCourse Fee: 12999 Rs.")
      case "Big Data Development" => println("You have choosen Big Data Development course!\nCourse Fee:
      case "Full Stack Development" => println("You have choosen Full Stack Development course!\nCourse Fe
      case "Spark" => println("You have choosen Spark course!\nCourse Fee: 19999 Rs.")
      case _ => println("You have choosen a different course which is not in our list!")
    }
  }
}
```

```
<terminated> MatchCaseDemo$ [Scala Application] /usr/lib/jvm/java-1.7.0-openjdk-1.7.0.79.x86_64/jre/bin/java (Jan 28, 2018, 6:58:16 PM)
You have choosen Spark course!
Course Fee: 19999 Rs.
```

Default output screenshot:



Scala - ListDemoProject/src/org/lists/MatchCaseDemo.scala - Eclipse
File Edit Refactor Navigate Search Project Scala Run Window Help
Quick Access
Java EE
Scala
MatchCaseDemo.scala
Console

```scala
package org.lists

object MatchCaseDemo {

  def main(args: Array[String]) {
    val course = "Network Security"
    course match {
      case "Android" => println("You have choosen Android course!\nCourse Fee: 12999 Rs.")
      case "Big Data Development" => println("You have choosen Big Data Development course!\nCourse Fee:
      case "Full Stack Development" => println("You have choosen Full Stack Development course!\nCourse Fe
      case "Spark" => println("You have choosen Spark course!\nCourse Fee: 19999 Rs.")
      case _ => println("You have choosen a different course which is not in our list!")
    }
  }
}
```

```
<terminated> MatchCaseDemo$ [Scala Application] /usr/lib/jvm/java-1.7.0-openjdk-1.7.0.79.x86_64/jre/bin/java (Jan 28, 2018, 7:03:05 PM)
You have choosen a different course which is not in our list!
```