

Bigdata Assignment 5.6

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.

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

Solution -

3. PartialFunction.scala

```
object PartialFunction { //partial function for sum
```

```
  def partialFunc(a:Int,b:Int):Int =
```

```
  {
```

```
    a+b+10
```

```
  }
```

```
//method for square
```

```
  def square(input:Int):Int =
```

```
  {
```

```
    input * input
```

```
  }
```

```
  def main(args:Array[String]) =
```

```
  {
```

```
    println("The Addition using parital function is : "
```

```
+partialFunc(args(0).toInt,args(1).toInt))
```

```
//partial function is passed to a method
```

```
    println("The Square using parital function Input is : "
```

```
+square(partialFunc(args(0).toInt,args(1).toInt)))
```

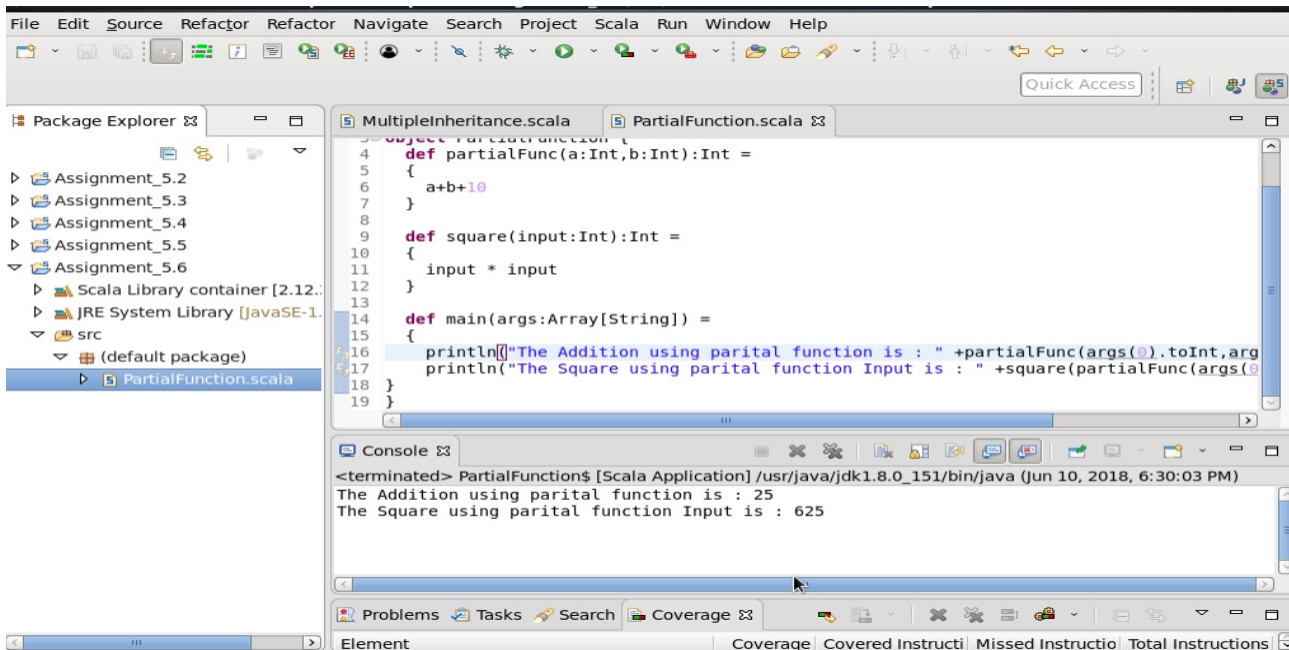
```
  }
```

```
}
```

INPUT – 5 10

OUTPUT -

We gave the inputs as 5 and 10 using the partial function we got the output 25 . And its square value we got 625 when we passed the partial function to a method. The output is shown in the console window of the below screenshot.



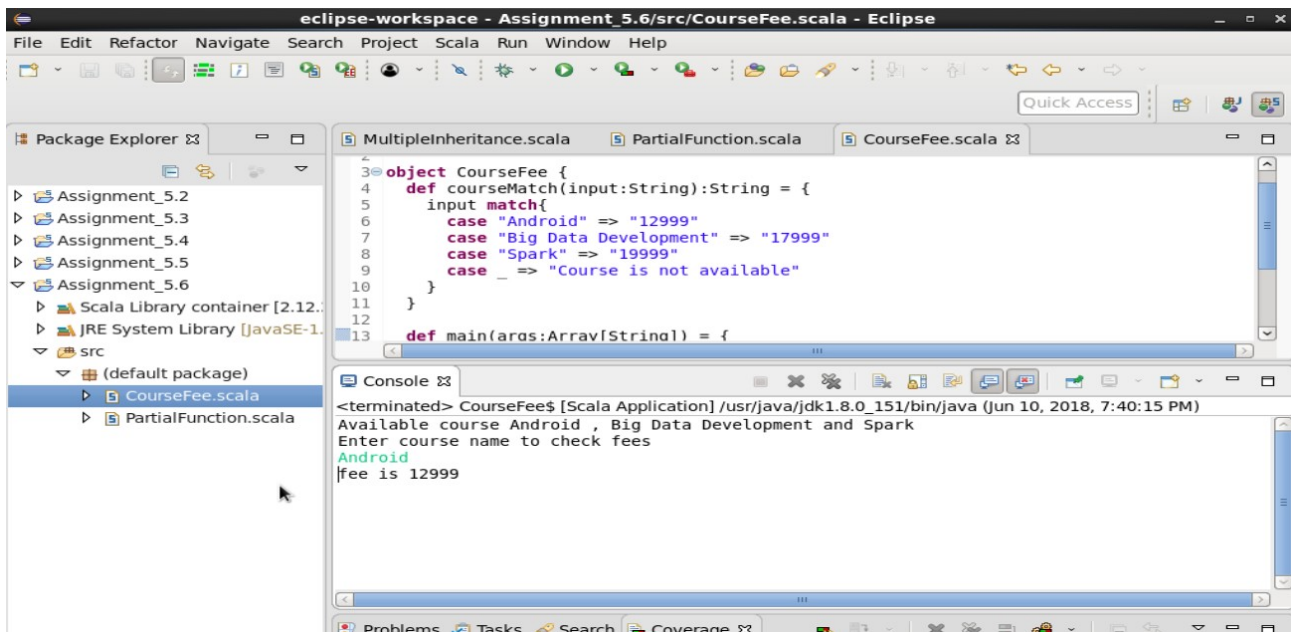
4 . CourseFee.scala

```
object CourseFee {  
  def courseMatch(input:String):String = {  
    //course matching function  
    input match {  
      case "Android" => "12999"  
      case "Big Data Development" => "17999"  
      case "Spark" => "19999"  
      case _ => "Course is not available"  
    }  
  }  
  
  def main(args:Array[String]) = {  
    println("Available course Android , Big Data Development and Spark")  
    println("Enter course name to check fees")  
  
    var course = scala.io.StdIn.readLine()//for string input  
    println("fee is "+courseMatch(course))  
  }  
}
```

Input – Android

Output – Fee is 12999.

As in the below screenshot we have the input in console window and we got the output. Whichever course name is given as input, it is matched with its corresponding fee.



The screenshot displays the Eclipse IDE interface. The Package Explorer on the left shows the project structure with 'Assignment_5.6' expanded, containing 'CourseFee.scala' and 'PartialFunction.scala'. The main editor window shows the code for 'CourseFee.scala':

```
3 object CourseFee {  
4   def courseMatch(input:String):String = {  
5     input match{  
6       case "Android" => "12999"  
7       case "Big Data Development" => "17999"  
8       case "Spark" => "19999"  
9       case _ => "Course is not available"  
10    }  
11  }  
12  
13  def main(args:Array[String]) = {
```

The Console window at the bottom shows the execution output:

```
<terminated> CourseFee$ [Scala Application] /usr/java/jdk1.8.0_151/bin/java (Jun 10, 2018, 7:40:15 PM)  
Available course Android , Big Data Development and Spark  
Enter course name to check fees  
Android  
fee is 12999
```