Assignment 15.2

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.

Code:

```
class ProgramPartialFunc {
  def squareFunc(n: Int) {
    println("Square: " + n * n)

  }
  def adder(m: Int, n: Int, p: Int) = m + n + p
  def partialFunc(x:Int,y:Int) {
    val add = adder( _: Int,20, _: Int)
    println("------------------------")
    println("Addition: "+add(x, y))
    squareFunc(add(x, y))
    println("------------------------")
  }
}

object PartialFunc_Task3 {

  def main(args: Array[String]) {

    println("Enter numbers")

    var x: Int = scala.io.StdIn.readLine().toInt;

    var y: Int = scala.io.StdIn.readLine().toInt;

    new ProgramPartialFunc().partialFunc(x, y)
  }
}
```

```scala
class ProgramPartialFunc {
  def squareFunc(n: Int) {
    println("Square: " + n * n)

  }
  def adder(m: Int, n: Int, p: Int) = m + n + p
  def partialFunc(x:Int,y:Int) {
    val add = adder( _: Int,20, _: Int)
    println("------------------------")
    println("Addition: "+add(x, y))
    squareFunc(add(x, y))
    println("------------------------")
  }
}


object PartialFunc_Task3 {

  def main(args: Array[String]) {

    println("Enter numbers")

    var x: Int = scala.io.StdIn.readLine().toInt;

    var y: Int = scala.io.StdIn.readLine().toInt;

    new ProgramPartialFunc().partialFunc(x, y)
  }
}

```

OUTPUT:

```
Run  PartialFunc_Task3
"C:\Program Files\Java\jdk1.8.0_144\bin\java" ...
Enter numbers
10
30
------------------------
Addition: 60
Square: 3600
------------------------

Process finished with exit code 0
```

2. 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.

```
122
123  def result(x: String): String = x match {
124    case "Android" => ("Android is 12999/-")
125    case "Big Data Development" => ("Big Data Development is 17999")
126    case "Big Data Development" => ("Big Data Development is 17999")
127    case "Spark" => ("Spark is 19999")
128    case _ => ("Oops! The course is yet not available")
129  }
```

OUTPUT:

```
130
131  result("Android")
132  result("Spark")
133  result("Java")
134
135
```

```
129
130
131  res8: String = Android is 12999/-
132  res9: String = Spark is 19999
133  res10: String = Oops! The course is yet not available
134
```