## **Assignment 15.2**

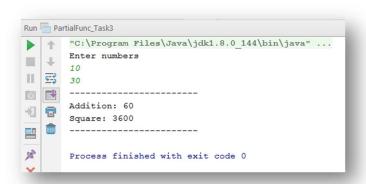
1. Write a partial function to add three numbers in which one number is constant and two numbers can be passed as inputs and define another method which can take the partial function as input and squares the result.

## 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)
 }
```

```
class ProgramPartialFunc {
        def squareFunc(n: Int) {
3
          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("---
14
17 Dobject PartialFunc_Task3 {
19
      def main(args: Array[String]) {
20
21
          println("Enter numbers")
          var x: Int = scala.io.StdIn.readLine().toInt;
24
        var y: Int = scala.io.StdIn.readLine().toInt;
26
27
           new ProgramPartialFunc().partialFunc(x, y)
28
29
```

## **OUTPUT:**



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.

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

## **OUTPUT:**