

# BIG DATA HADOOP AND SPARK DEVELOPMENT

## ASSIGNMENT 17

### Table of Contents:

1. Introduction	2
2. Objective	2
3. Problem Statement	2
4. Expected Output	
• Task 1	3
• Task 2	4
• Task 3	5
• Task 4	6

# BIG DATA HADOOP AND SPARK DEVELOPMENT

## 1. Introduction

In this assignment, the given tasks are performed and Output of the tasks are recorded in the form of Screenshots.

## 2. Objective

This Assignment consolidates the deeper understanding of the Session – 17 SCALA BASICS 4

## 3. Problem Statement

- Task 1
  - Write a simple program to show inheritance in scala.
- Task 2
  - Write a simple program to show multiple inheritance in scala
- 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.
- 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.

## Expected Output

- Task 1
  - Write a simple program to show inheritance in scala.

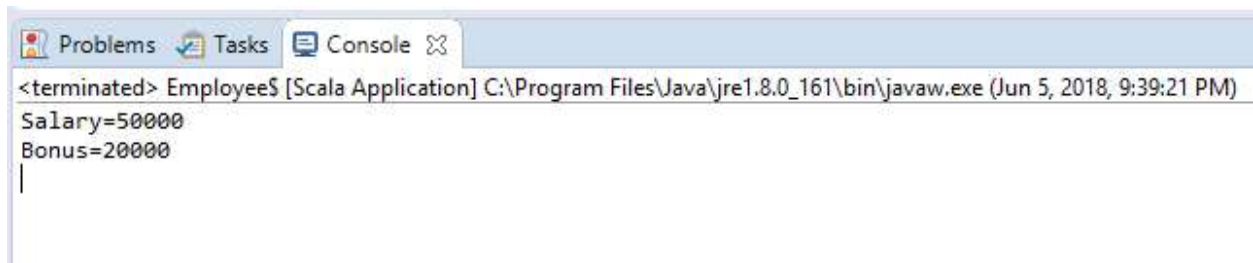
Following program is the example of inheritance in scala

```
class Performance{  
    val Salary:Int = 50000  
}  
class Aprisal extends Performance{  
    val bonus: Int =20000  
  
    println("Salary="+Salary)  
    println("Bonus="+bonus)  
}  
  
object Employee {  
    def main(args:Array[String]){  
        new Aprisal()  
    }  
}
```

Output of the above program is shown below

```
Salary=50000  
Bonus=20000
```

Screenshot of the output of the program inheritance



- Task 2
- Write a simple program to show multiple inheritance in scala

Below is the screen shot of the example program to show the multiple inheritance in scala

```
class Performance{  
    val Salary:Int = 50000  
}  
class Aprisal extends Performance{  
    val bonus: Int =20000  
    println("Salary="+Salary)  
    println("Bonus="+bonus)  
}  
class total extends Aprisal {  
    val Total= Salary+bonus  
    println("Total="+Total)  
}  
object Employee {  
    def main(args:Array[String]){  
        new total()  
    }  
}
```

Output of the above program is shown below



The screenshot shows a Java IDE interface with a tab labeled 'Console'. The console output displays the results of the Scala program execution, showing the salary, bonus, and total values.

```
<terminated> Employee$ [Scala Application] C:\Program Files\Java\jre1.8.0_161\bin\javaw.exe (Jun 5, 2018, 10:02:12 PM)  
Salary=50000  
Bonus=20000  
Total=70000
```

- 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.

Following is the program to create a partial function.

A partial function is written to add three numbers in which one number is constant and two numbers are passed as inputs and another is defined as method which takes the partial function as input and squares the result.

Constant is 10 in the program

PartialFn.scala

```
class PartialFn {  
    val addcon: PartialFunction[(Int, Int), Int] = {  
        case (a, b) => a + b + 10  
    }  
    def square(x: Int) = x * x  
}  
object PartialFn {  
    def main(arg: Array[String]) {  
        val pf = new PartialFn()  
        println("Sum of three numbers: " + pf.addcon(1, 2))  
        val result = pf.square(pf.addcon(1, 2))  
        println("Final Square: " + result)  
    }  
}
```

Output of the program is given below.

Problems Tasks Console

<terminated> PartialFn\$ [Scala Application] C:\Program Files\Java\jre1.8.0\_161\bin\javaw.exe (Jun 5, 2018, 11:26:30 PM)

Sum of three numbers: 13

Final Square: 169

## • 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.

```
matching.scala

class matching {

  def matchTest(x: String): String = x match {
    case "Android App Development" => "14,999 INR"
    case "Data Science" => "49,999 INR"
    case "Big Data Hadoop & Spark Developer" => "24,999 INR"
    case "Blockchain Certification" => "49,999 INR"
    case _ => "Please enter relevant course name"
  }
}

object MatchinDemo {
  def main(args: Array[String]){
    val m = new matching()

    println("Course Name: " + "Android App Development" + "\nCourse Fee: "+m.matchTest("Android App Development"))
    println("Course Name: " + "Data Science" + "\nCourse Fee: "+m.matchTest("Data Science"))
    println("Course Name: " + "Big Data Hadoop & Spark Developer" + "\nCourse Fee: "+m.matchTest("Big Data Hadoop & Spark Developer"))
    println("Course Name: " + "Blockchain Certification" + "\nCourse Fee: "+m.matchTest("Blockchain Certification"))
    println("Course Name: " + "Java Concepts" + "\nCourse Fee: "+m.matchTest("Java Concepts"))
  }
}
```

```
class matching {

  def matchTest(x: String): String = x match {
    case "Android App Development" => "14,999 INR"
    case "Data Science" => "49,999 INR"
    case "Big Data Hadoop & Spark Developer" => "24,999 INR"
    case "Blockchain Certification" => "49,999 INR"
    case _ => "Please enter relevant course name"
  }
}

object MatchinDemo {
  def main(args: Array[String]){
    val m = new matching()

    println("Course Name: " + "Android App Development" + "\nCourse Fee: "
    +m.matchTest("Android App Development"))
    println("Course Name: " + "Data Science" + "\nCourse Fee: " +m.matchTest("Data Science"))
    println("Course Name: " + "Big Data Hadoop & Spark Developer" + "\nCourse Fee: "
    +m.matchTest("Big Data Hadoop & Spark Developer"))
    println("Course Name: " + "Blockchain Certification" + "\nCourse Fee: "
    +m.matchTest("Blockchain Certification"))
    println("Course Name: " + "Java Concepts" + "\nCourse Fee: "
    +m.matchTest("Java Concepts"))
  }
}
```

## Output of the program



The screenshot shows an IDE interface with three tabs: 'Problems', 'Tasks', and 'Console'. The 'Console' tab is active, displaying the output of a Java application. The output starts with a terminated status and the file path. It then lists several course details: 'Android App Development' with a fee of 14,999 INR, 'Data Science' with a fee of 49,999 INR, 'Big Data Hadoop & Spark Developer' with a fee of 24,999 INR, and 'Blockchain Certification' with a fee of 49,999 INR. The program then prints 'Java Concepts' and prompts the user to enter a course name.

```
<terminated> MatchinDemo$ [Scala Application] C:\Program Files\Java\jre1.8.0_161\bin\javaw.exe (Jun 5, 2018, 11:43:32 PM)
Course Name: Android App Development
Course Fee: 14,999 INR
Course Name: Data Science
Course Fee: 49,999 INR
Course Name: Big Data Hadoop & Spark Developer
Course Fee: 24,999 INR
Course Name: Blockchain Certification
Course Fee: 49,999 INR
Course Name: Java Concepts
Course Fee: Please enter relavent course name
```