

## Assignment - 17.1

**Task1:** Write a simple program to show inheritance in scala.

**Answer:**

Here we are inheriting Employee class in Programmer class and then getting details.

```
Employee.scala
```

```
class Employee{
    val salary:Float = 10000
}

class Programmer extends Employee{
    val bonus:Int = 5000
    println("Salary = "+salary)
    println("Bonus = "+bonus)
}

object Employee {
    def main(args: Array[String]){
        new Programmer()
    }
}
```

**Output:**

```
Problems Tasks Console Git Staging
<terminated> Employee$ [Scala Application] C:\Program Files\Java\jre1.8.0_171\bin\javaw.exe (12-May-2018, 1:00:14 PM)
Salary = 10000.0
Bonus = 5000
```

**Task2:** Write a simple program to show multiple inheritances in scala.

**Answer:**

Here we are inheriting Employee class in Programmer class and then getting details.

```
*Employee.scala

class Employee{
    val salary:Float = 10000
}

class Programmer extends Employee{
    val bonus:Int = 5000
    println("Salary = "+salary)
    println("Bonus = "+bonus)
}

class FinalSummary extends Programmer{

    val totalSalary = salary + bonus
    println("Total Salary = " +totalSalary)
}

object Employee {
    def main(args: Array[String]){

        val finalSal = new FinalSummary()

    }
}
```

### **Output:**

```
Problems Tasks Console Git Staging
<terminated> Employee$ [Scala Application] C:\Program Files\Java\jre1.8.0_171\bin\javaw.exe (12-May-2018, 1:19:22 PM)
Salary = 10000.0
Bonus = 5000
Total Salary = 15000.0
```

**Task3:** 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.

### **Answer:**

First we are getting the sum of three numbers using partial function in Scala and then we have defined the Square function to get the square of the previous result.

```
partialFunctionDemo.scala Employee.scala LearningScala3.sc

class partialFunction{
  val addConstantTo: PartialFunction[(Int, Int), Int] = {
    case (a, b) => a + b + 5
  }

  def squareIt(x: Int) = x*x
}

object partialFunctionDemo {
  def main(args: Array[String]){
    val pf = new partialFunction()

    println("Sum of three numbers: "+pf.addConstantTo(1,2))

    val res = pf.squareIt(pf.addConstantTo(1,2))
    println("Final Square: "+res)
  }
}
```

#### **Output:**

```
Problems Tasks Console Git Staging
<terminated> partialFunctionDemo$ [Scala Application] C:\Program Files\Java\jre1.8.0_171\bin\javaw.exe (12-May-2018, 11:00:56 PM)
Sum of three numbers: 8
Final Square: 64
|
```

**Task4:** 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.

#### **Answer:**

First we are defining the course fees for each of the course name mentioned in the above list and a default message is set if any other course is entered to get the fees.

I have got output for all possibilities of inputs. Please find the screenshots below.

```

MatchinDemo.scala
class MatchinDemo{
  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 relavent course name"
  }
}

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

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

```

### Output:

```

Problems Tasks Console Git Staging
<terminated> MatchinDemo$ [Scala Application] C:\Program Files\Java\jre1.8.0_171\bin\javaw.exe (13-May-2018, 12:17:08 AM)
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

```

### Code Snippet:

```

class MatchinDemo{

  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 relavent course name"
  }
}

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

    println("Course Name: " + "Android App Development" + "\nCourse Fee: "+md.matchTest("Android App Development"))
    println("Course Name: " + "Data Science" + "\nCourse Fee: "+md.matchTest("Data Science"))
    println("Course Name: " + "Big Data Hadoop & Spark Developer" + "\nCourse Fee: "+md.matchTest("Big Data Hadoop & Spark Developer"))
  }
}

```

```
        println("Course Name: " + "Blockchain Certification" + "\nCourse Fee: " + md.matchTest("Blockchain Certification"))
        println("Course Name: " + "Java Concepts" + "\nCourse Fee: " + md.matchTest("Java Concepts"))
    }
}
```

**Task4:** 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.

**Answer:**

First we are defining the course fees for each of the course name mentioned in the above list and a