Scala 4:
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.

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

```
class Employee{
        var sal:Float = 1000
    }


class TotalSal extends Employee{
       var bonus:Int = 5000
      def x()
     {
      println("Salary = "+sal)
       println("Bonus = "+bonus)
     }
     }


 object Testbject{
       def main(args:Array[String]){
           val obj =  new TotalSal()
     obj.x
        }
     }
```

```
class Employee{
          var sal:Float = 1000
      }

      class TotalSal extends Employee{
        var bonus:Int = 5000
        def x()
        {
         println("Salary = "+sal)
         println("Bonus = "+bonus)
        }
      }


    object tsk1 {
          def main(args:Array[String]){
             val obj =  new TotalSal()
            obj.x
          }
        }
```

Task 2:

Write a simple program to show multiple inheritance in scala

```
class Basecls{
      def a()
        {
        println("version 1")
        }
   }



class dervie1cls extends Basecls{
      def b()
        {
        println("version 2")
        }
```

```
    }



class inhertiancecls extends dervie1cls{
        def c()
           {
           println("version 2 and version 2 using multiple inheritance")
           }
    }

object Inherit
{
  def main(args: Array[String])
  {
           val obj = new inhertiancecls
           obj.a
           obj.b
           obj.c
  }
}
```

```
class Basecls{
          def a()
  {
  println("version 1")
  }
      }



class dervie1cls extends Basecls{
          def b()
  {
  println("version 2")
  }
      }



class inhertiancecls extends dervie1cls{
          def c()
  {
  println("version 2 and version 2 using multiple inheritance")
  }
      }

object Inherit
{
  def main(args: Array[String])
  {
  val obj = new inhertiancecls
  obj.a
  obj.b
  obj.c
  }
}
```

```
scala> class Basecls{
     |           def a()
     |   {
     |   println("version 1")
     |   }
     |       }
defined class Basecls

scala> class dervie1cls extends Basecls{
     |           def b()
     |   {
     |   println("version 2")
     |   }
     |       }
defined class dervie1cls

scala> class inhertiancecls extends dervie1cls{
     |           def c()
     |   {
     |   println("version 2 and version 2 using multiple inheritance")
     |   }
     |       }
defined class inhertiancecls

scala> object Inherit
defined object Inherit

scala> {
     |   def main(args: Array[String])
     |   {
     |   val obj = new inhertiancecls
     |   obj.a
     |   obj.b
     |   obj.c
     |   }
     | }
```

Task 3:

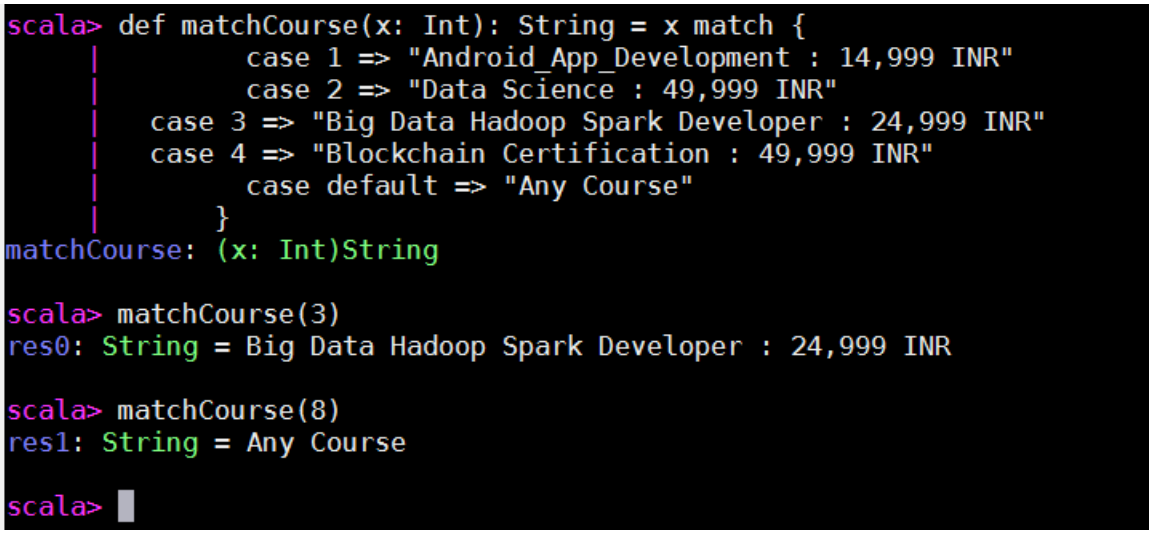
Write a partial function to add three numbers in which one number is constant and two numbers can be passed as inputs and define another method which can take the partial function as input and squares the result.

```
object MainObject {
  val addConstantTo: PartialFunction[(Int, Int), Int] = {
        case (a, b) => a + b + 1
      }

   def main(args: Array[String]) = {
     var sum = addConstantTo(10,10)      // Function composition

     var x: Double = sum
```

```
    var result = x * x

    println(result)
  }
 }

object MainObject {
  val addConstantTo: PartialFunction[(Int, Int), Int] = {
        case (a, b) => a + b + 1
      }

   def main(args: Array[String]) = {
    var sum = addConstantTo(10,10)      // Function composition

    var x: Double = sum

    var result = x * x

    println(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.

```
def matchCourse(x: Int): String = x match {
     case 1 => "Android_App_Development : 14,999 INR"
     case 2 => "Data Science : 49,999 INR"
         case 3 => "Big Data Hadoop Spark Developer : 24,999 INR"
         case 4 => "Blockchain Certification : 49,999 INR"
     case default => "Any Course"
   }
```


scala> def matchCourse(x: Int): String = x match {
case 1 => "Android_App_Development : 14,999 INR"
case 2 => "Data Science : 49,999 INR"
case 3 => "Big Data Hadoop Spark Developer : 24,999 INR"
case 4 => "Blockchain Certification : 49,999 INR"
case default => "Any Course"
}
matchCourse: (x: Int)String
scala> matchCourse(3)
res0: String = Big Data Hadoop Spark Developer : 24,999 INR
scala> matchCourse(8)
res1: String = Any Course
scala>