Task 1

Write a simple program to show inheritance in scala.

Code-

```
object Inheritance {

  class Person {
    var Persionid:Int=1001
    def get_persionid(){
      println("Persion id:"+Persionid)
    }
  }
  class Student extends Person {
    var Studentid:String="2x007"
    def get_Studentid() {
      get_persionid()
      println("Student id:"+Studentid)
    }
  }

  def main(args:Array[String]){
    new Student().get_Studentid()
  }
}
```

Output- class Student is inheriting the attributes and method of class Person

```
  class Student extends Person {
    var Studentid:String="2x007"
    def get_Studentid() {
      get_persionid()
      println("Student id:"+Studentid)
    }
  }

  def main(args:Array[String]){
    new Student().get_Studentid()
  }
}
```

Problems Tasks Console Scala Interpreter (Scala_Project1)

```
<terminated> Inheritance$ [Scala Application] C:\Program Files\Java\jre
Persion id:1001
Student id:2x007
```

Task 2

Write a simple program to show multiple inheritance in scala

Code-

```
object Multiple_Inheritance {
       def main(args: Array[String]): Unit= {

    trait A {
      var distance: Int = _
      def action = {
        distance = distance + 5
      }
    }

    trait B {
      var driverVar: Int = _
      def action = {
        driverVar = driverVar + 1
      }
    }

    class AB extends A with B {
     distance = 3;
     driverVar = 6;
     override def action = {
      super[A].action
      super[B].action
     }
    }

    var ab = new AB
    ab.action
    println(ab.driverVar)
    println(ab.distance)
  }
}
```

Output-

```
    var ab = new AB
    ab.action
    println(ab.driverVar)
    println(ab.distance)
  }
}
```

Problems Tasks Console Scala Interpreter (Scala_Project1)

```
<terminated> Inheritance$ [Scala Application] C:\Program Files\Java\jre1.8.0_191\b
Persion id:1001
Student id:2x007
```

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.

Code-

```
object Partial_Functions {

  def main(args:Array[String]){

  def add(x:Int,y:Int,z:Int)=x+y+z
  val add_2number=add(4,_:Int,_:Int)
  println(add_2number(2,3))

  def square(x:Int):Int=x*x
  println(square(add_2number(2,3)))
  }
}
```

Output-

```
  println(add_2number(2,3))

  def square(x:Int):Int=x*x
  println(square(add_2number(2,3)))
  }
}
```

Problems Tasks Console Scala Interpreter (Scala_Pr

```
<terminated> Partial_Functions$ [Scala Application] C:\Program F
9
81
```

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.

Code:

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

  println("Enter course name:")
  var input=scala.io.StdIn.readLine()
  var price= input match {
      case "Android App Development" =>println("14,999 INR")
      case "Data Science"=>println("49,999 INR")
      case "Big Data Hadoop & Spark Developer "=>println("24,999 INR")
      case "Blockchain Certification"=>println("49,999 INR")
      case _=>println("price not available")
  }
  }
}
```

Output-

```
<terminated> Match$ [Scala Application] C:\Program Files\Java\ji
Enter course name:
Data Science
49,999 INR
```

```
Problems  Tasks  Console  Scala Interpreter (S
<terminated> Match$ [Scala Application] C:\Program File
Enter course name:
Java Certification
price not available
```