

Assignment – 17.1 Scala 4

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

Program:

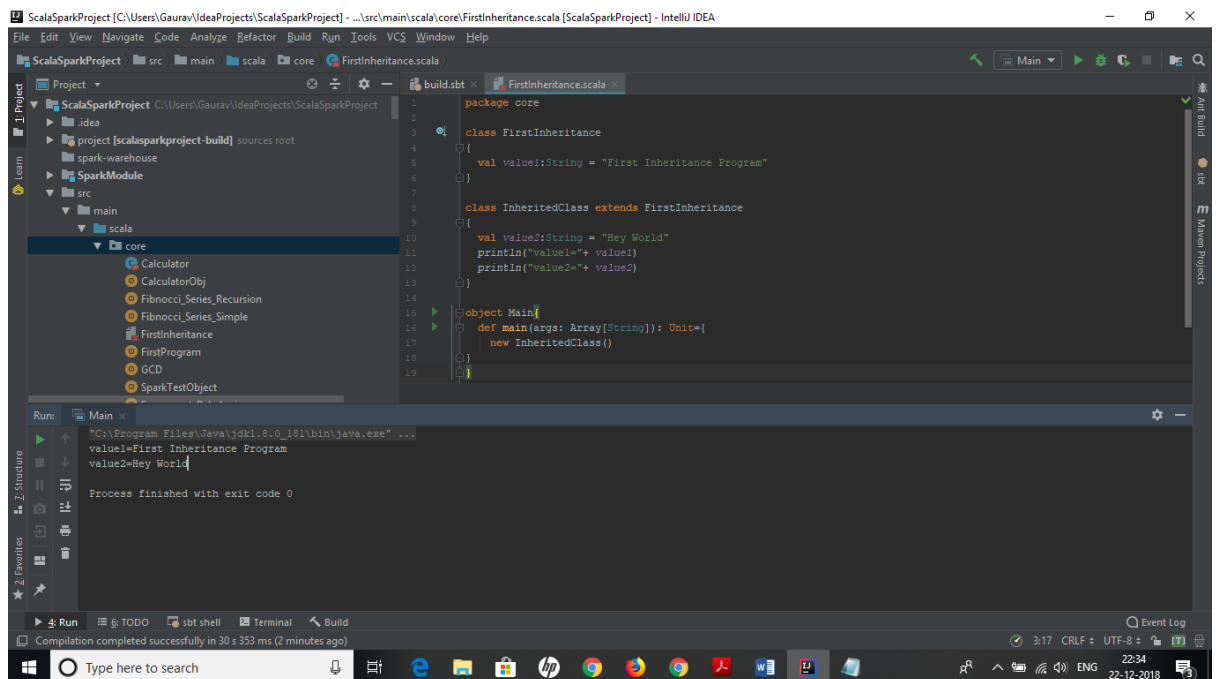
```
package core

class FirstInheritance
{
    val value1:String = "First Inheritance Program"
}

class InheritedClass extends FirstInheritance
{
    val value2:String = "Hey World"
    println("value1="+ value1)
    println("value2="+ value2)
}

object Main{
    def main(args: Array[String]): Unit={
        new InheritedClass()
    }
}
```

Output:

The screenshot shows the IntelliJ IDEA IDE with a Scala project named 'ScalaSparkProject'. The project structure on the left includes 'src/main/scala/core'. The main editor displays the 'FirstInheritance.scala' file with the code for 'FirstInheritance', 'InheritedClass', and 'Main'. The 'Run' tab at the bottom shows the output of the program: 'value1=First Inheritance Program' and 'value2=Hey World', followed by 'Process finished with exit code 0'. The status bar at the bottom indicates 'Compilation completed successfully in 30 s 353 ms (2 minutes ago)'.

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

Program:

```

package core

trait MultipleInheritance
{
  def show()
  {
    println("Assignment 17.1")
  }
}

trait one extends MultipleInheritance
{
  override def show()
  {
    println("Multiple Inheritance")
  }
}

trait two extends MultipleInheritance
{
  override def show()
  {
    println("This will be printed: Hi Scala")
  }
}

class three extends one with two

object MainMulti {
  def main(args: Array[String]): Unit = {
    var c: three = new three
    c.show()
  }
}

```

Output:

The screenshot shows the IntelliJ IDEA IDE with the ScalaSparkProject open. The project structure on the left includes a 'core' package with several files. The 'MultipleInheritance.scala' file is open in the editor, displaying the code from the previous block. The 'Run' tab at the bottom shows the output of the 'MainMulti' object, which is 'This will be printed: Hi Scala'. The status bar at the bottom indicates that the compilation was successful.

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.

Program:

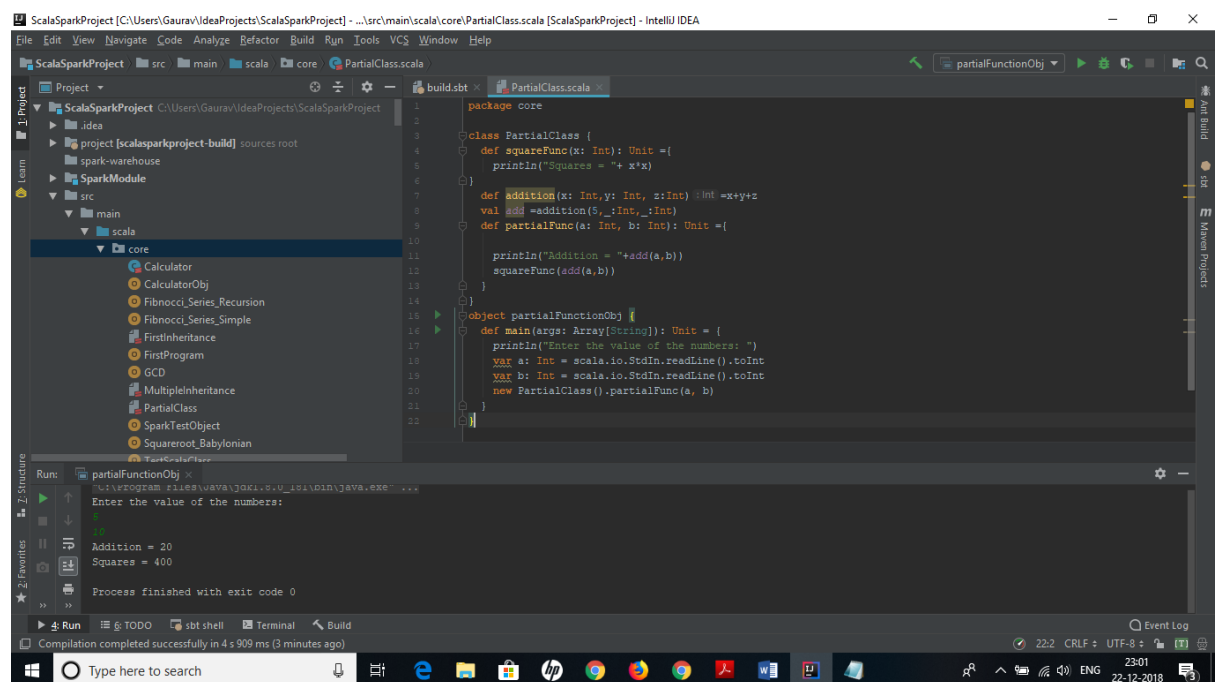
```
package core

class PartialClass {
  def squareFunc(x: Int): Unit = {
    println("Squares = " + x*x)
  }
  def addition(x: Int, y: Int, z: Int) = x + y + z
  val add = addition(5, _: Int, _: Int)
  def partialFunc(a: Int, b: Int): Unit = {

    println("Addition = " + add(a, b))
    squareFunc(add(a, b))
  }
}

object partialFunctionObj {
  def main(args: Array[String]): Unit = {
    println("Enter the value of the numbers: ")
    var a: Int = scala.io.StdIn.readLine().toInt
    var b: Int = scala.io.StdIn.readLine().toInt
    new PartialClass().partialFunc(a, b)
  }
}
```

Output:



The screenshot shows the IntelliJ IDEA IDE with a Scala project named 'ScalaSparkProject'. The code editor displays the same Scala code as shown in the previous block. The 'Run' tab at the bottom shows the execution output for the 'partialFunctionObj' object. The output is as follows:

```
Run: partialFunctionObj
Enter the value of the numbers:
20
400
Addition = 20
Squares = 400
Process finished with exit code 0
```

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.

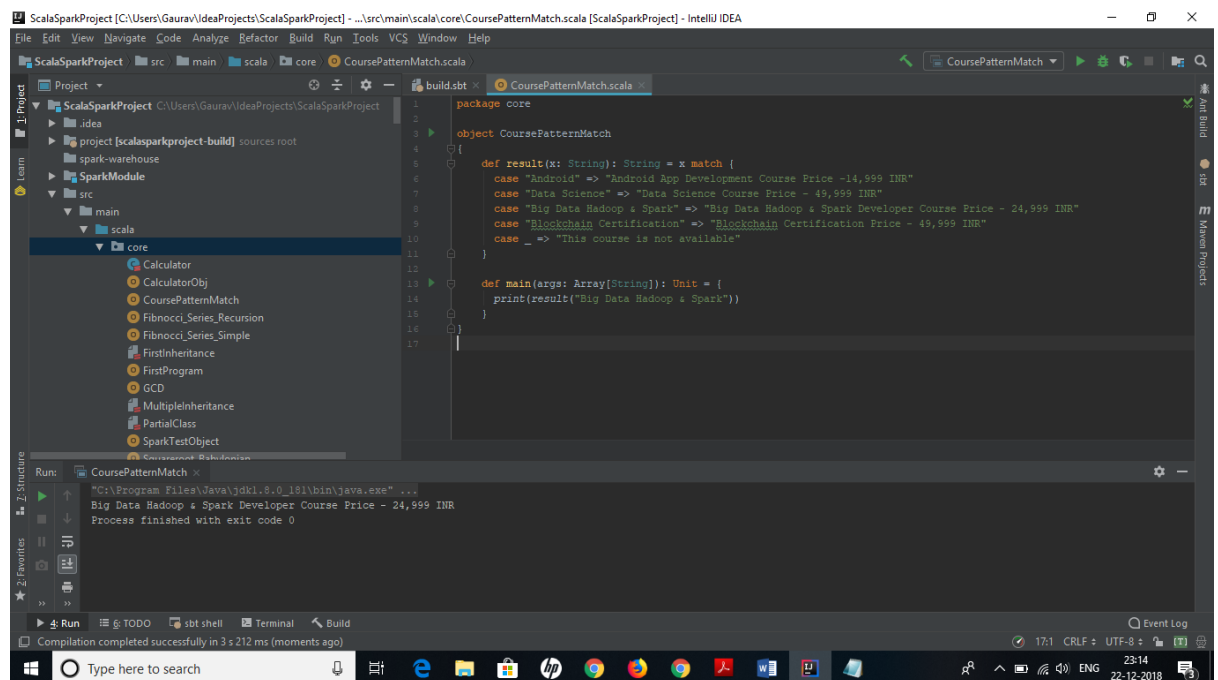
Program:

```
package core

object CoursePatternMatch
{
    def result(x: String): String = x match {
        case "Android" => ("Android App Development Course Price -14,999 INR")
        case "Data Science" => ("Data Science Course Price - 49,999 INR")
        case "Big Data Hadoop & Spark" => ("Big Data Hadoop & Spark Developer Course Price – 24,999 INR")
        case "Blockchain Certification" => ("Blockchain Certification Price – 49,999 INR")
        case _ => ("This course is not available")
    }

    def main(args: Array[String]): Unit = {
        print(result("Big Data Hadoop & Spark"))
    }
}
```

Output:



The screenshot shows the IntelliJ IDEA IDE with the ScalaSparkProject open. The project structure on the left includes a 'core' package containing several objects. The 'CoursePatternMatch.scala' file is selected, showing the Scala code for the match function and the main method. The 'Run' tab at the bottom displays the output of the program, which prints 'Big Data Hadoop & Spark Developer Course Price - 24,999 INR'.

```
ScalaSparkProject [C:\Users\Gaurav\IdeaProjects\ScalaSparkProject] - ...src\main\scala\core\CoursePatternMatch.scala [ScalaSparkProject] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
ScalaSparkProject src main scala core CoursePatternMatch.scala
Project
  ScalaSparkProject
    .idea
    project [scalasparkproject-build] sources root
    spark-warehouse
    SparkModule
      src
        main
          scala
            Calculator
            CalculatorObj
            CoursePatternMatch
            Fibonacci_Series_Recursion
            Fibonacci_Series_Simple
            FirstInheritance
            GCD
            MultipleInheritance
            PartialClass
            SparkTestObject
  build.sbt
  CoursePatternMatch.scala
1 package core
2
3 object CoursePatternMatch
4 {
5     def result(x: String): String = x match {
6         case "Android" => "Android App Development Course Price -14,999 INR"
7         case "Data Science" => "Data Science Course Price - 49,999 INR"
8         case "Big Data Hadoop & Spark" => "Big Data Hadoop & Spark Developer Course Price - 24,999 INR"
9         case "Blockchain Certification" => "Blockchain Certification Price - 49,999 INR"
10        case _ => "This course is not available"
11    }
12
13    def main(args: Array[String]): Unit = {
14        print(result("Big Data Hadoop & Spark"))
15    }
16
17
18 Run CoursePatternMatch
19 C:\Program Files\Java\jdk1.8.0_181\bin\java.exe ...
20 Big Data Hadoop & Spark Developer Course Price - 24,999 INR
21 Process finished with exit code 0
22
23 Run g: TODO sbt shell Terminal Build
24 Compilation completed successfully in 3 s 212 ms (moments ago)
25 Type here to search
26 17:1 CRLF UTF-8 ENG 23:14 22-12-2018
```

