

Session 17:

SCALA BASICS 4

Assignment 1

Task 1

Write a simple program to show inheritance in scala.

Class

```
package Inheritance

class Simple (speed : Int){
  val mph :Int = speed
  def race() = println("Racing")
}

class Car (speed : Int) extends Simple(speed) {
  override val mph: Int= speed
  override def race() = println("Racing Car")
}

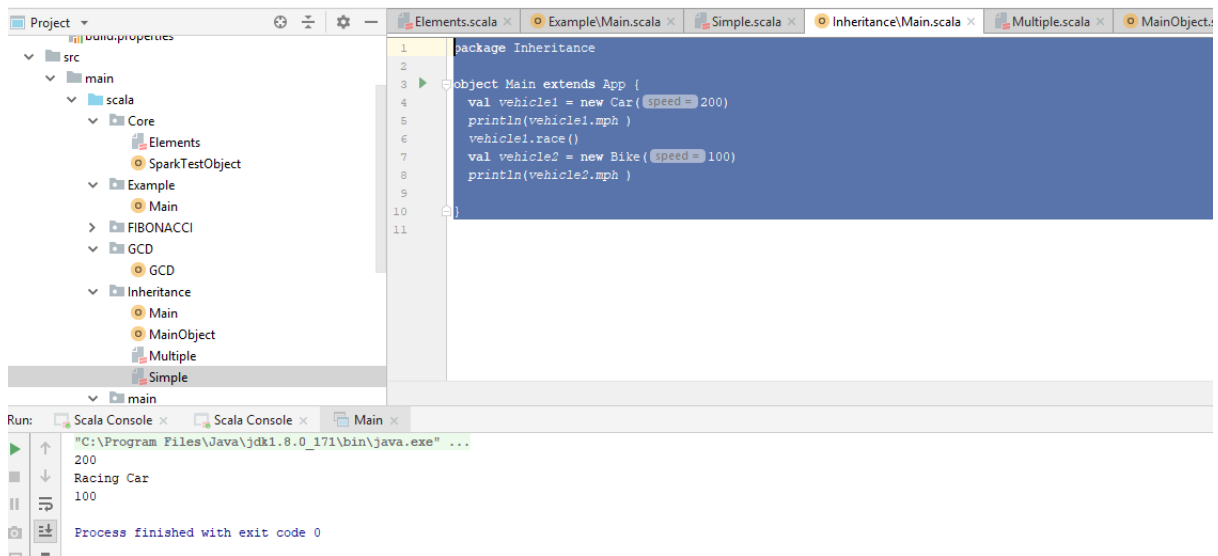
class Bike(speed : Int) extends Simple(speed) {
  override val mph: Int = speed
  override def race() = println("Racing Bike")
}
```

Object :

```
package Inheritance

object Main extends App {
  val vehicle1 = new Car(200)
  println(vehicle1.mph )
  vehicle1.race()
  val vehicle2 = new Bike(100)
  println(vehicle2.mph )
}
```

Output:



Task 2

Write a simple program to show multiple inheritance in scala

```

package Inheritance

class Multiple{
    var salary1 = 10000
}

class B extends Multiple{
    var salary2 = 20000
}

class C extends B{
    def show(){
        println("salary1 = "+salary1)
        println("salary2 = "+salary2)
    }
}

```

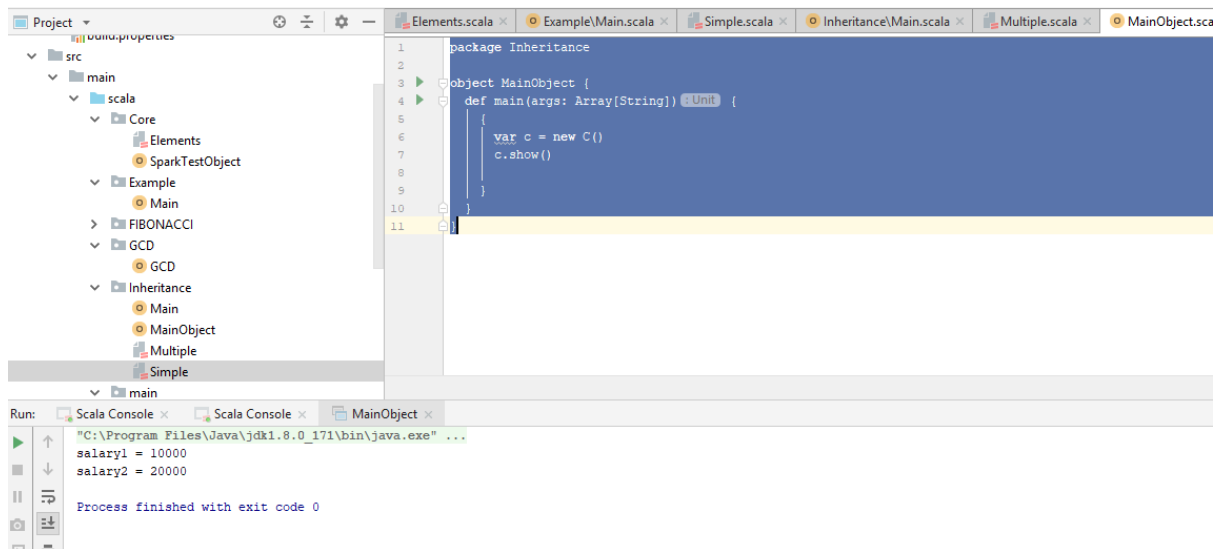
```

package Inheritance

object MainObject {
    def main(args: Array[String]) {
        {
            var c = new C()
            c.show()
        }
    }
}

```

Output:



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.

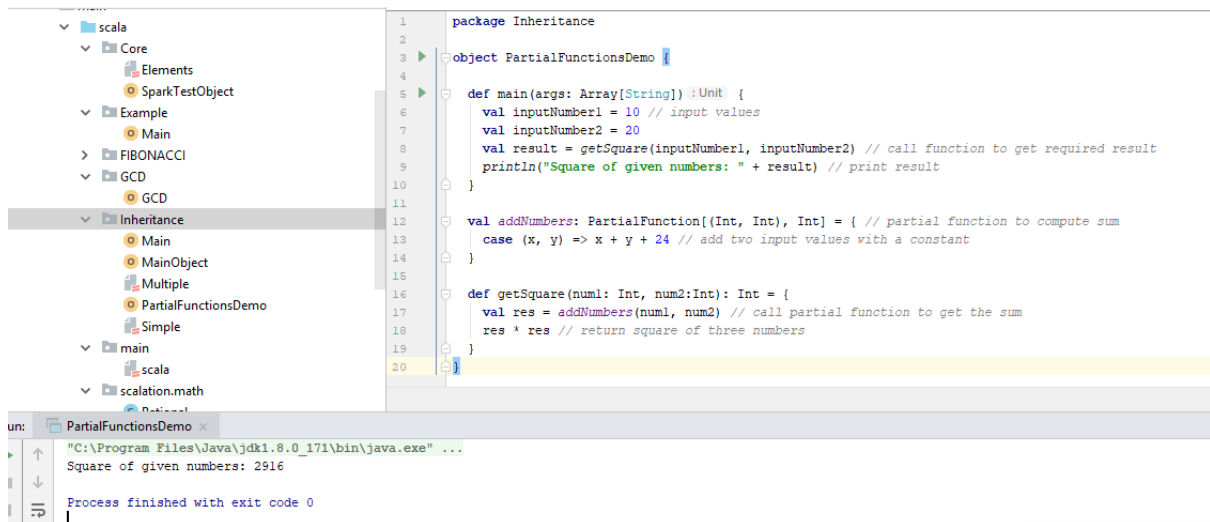
Solution:

```
object PartialFunctionsDemo {
```

```
  def main(args: Array[String]) {
    val inputNumber1 = 10 // input values
    val inputNumber2 = 20
    val result = getSquare(inputNumber1, inputNumber2) // call function to get required result
    println("Square of given numbers: " + result) // print result
  }
```

```
  val addNumbers: PartialFunction[(Int, Int), Int] = { // partial function to compute sum
    case (x, y) => x + y + 24 // add two input values with a constant
  }
```

```
  def getSquare(num1: Int, num2: Int): Int = {
    val res = addNumbers(num1, num2) // call partial function to get the sum
    res * res // return square of three numbers
  }
}
```



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 Code

```

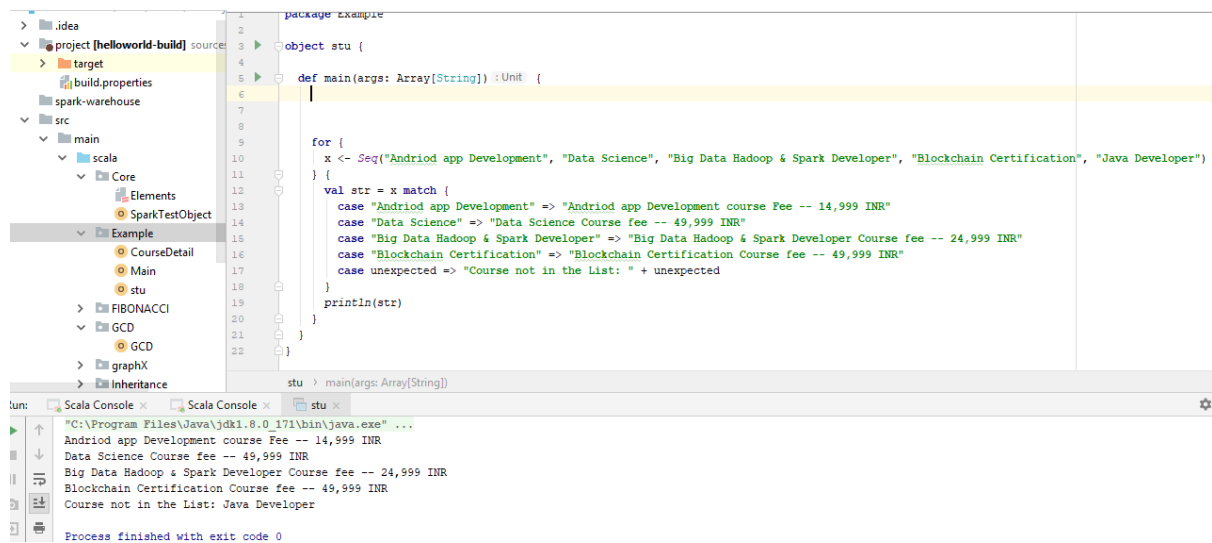
package Example

object stu {

    def main(args: Array[String]) {

        for {
            x <- Seq("Andriod app Development", "Data Science", "Big Data Hadoop & Spark Developer", "Blockchain Certification", "Java Developer")
        } {
            val str = x match {
                case "Andriod app Development" => "Andriod app Development course Fee -- 14,999 INR"
                case "Data Science" => "Data Science Course fee -- 49,999 INR"
                case "Big Data Hadoop & Spark Developer" => "Big Data Hadoop & Spark Developer Course fee -- 24,999 INR"
                case "Blockchain Certification" => "Blockchain Certification Course fee -- 49,999 INR"
                case unexpected => "Course not in the List: " + unexpected
            }
            println(str)
        }
    }
}

```



```
package Example
```

```
object CourseDetail {
```

```

  def main(args: Array[String]) {
    val Course1 = new Course("Andriod app Development","14000 INR")
    val Course2 = new Course("Data Science","49,999 INR")
    val Course3 = new Course("Big Data Hadoop & Spark Developer","24,999 INR")
    val Course4 = new Course("Blockchain Certification","49,999 INR")
    val Course5 = new Course("Java Advanced","10,999 INR")

    for (st <- List(Course1,Course2,Course3,Course4)) {

      st match {
        case Course("Andriod app Development","14000 INR") => println("Andriod app
Development course Fee -- 14,999 INR")
        case Course("Data Science","49,999 INR") => println("Data Science Course
fee -- 49,999 INR")
        case Course("Big Data Hadoop & Spark Developer","24,999 INR") =>
println("Big Data Hadoop & Spark Developer Course fee -- 24,999 INR")
        case Course("Blockchain Certification","49,999 INR") => println("Blockchain
Certification Course fee -- 49,999 INR")
        case Course("unexpected","not in list") => println("Course not in the
List: :")
        // case Course("Blockchain Certification","49,999 INR") =>
println("Blockchain Certification Course fee -- 49,999 INR")
        case Course(coursename,price) =>
println("Course: " + coursename + " Price: " + price )
      }
    }
  }

  case class Course(coursename:String, price: String)
}

```

The screenshot shows an IDE with a project named "helloworld-build" and a Scala file named "CourseDetail.scala". The code defines a `Course` class and a `main` function that prints course details for a list of courses.

```
10 val Course1 = new Course("Android app Development", "14000 INR")
11 val Course2 = new Course("Data Science", "49,999 INR")
12 val Course3 = new Course("Big Data Hadoop & Spark Developer", "24,999 INR")
13 val Course4 = new Course("Blockchain Certification", "49,999 INR")
14 val Course5 = new Course("Java Advanced", "10,999 INR")
15
16
17 for (st <- List(Course1, Course2, Course3, Course4)) {
18   st match {
19     case Course("Android app Development", "14000 INR") => println("Android app Development course Fee -- 14,999 INR")
20     case Course("Data Science", "49,999 INR") => println("Data Science Course fee -- 49,999 INR")
21     case Course("Big Data Hadoop & Spark Developer", "24,999 INR") => println("Big Data Hadoop & Spark Developer Course fee -- 24,999 INR")
22     case Course("Blockchain Certification", "49,999 INR") => println("Blockchain Certification Course fee -- 49,999 INR")
23     case Course("unexpected", "not in list") => println("Course not in the List: ")
24     // case Course("Blockchain Certification", "49,999 INR") => println("Blockchain Certification Course fee -- 49,999 INR")
25     case Course(courseName, price) =>
26       | println("Course: " + courseName + " Price: " + price )
27   }
28 }
29
30 case class Course(courseName:String, price: String)
31
```

The output of the program is shown in the Scala Console:

```
un: C:\Program Files\Java\jdk1.8.0_171\bin\java.exe ...
Android app Development course Fee -- 14,999 INR
Data Science Course fee -- 49,999 INR
Big Data Hadoop & Spark Developer Course fee -- 24,999 INR
Blockchain Certification Course fee -- 49,999 INR
Process finished with exit code 0
```