#### **Problem Statement-**

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

#### **Solution-**

To demonstrate simple inheritance in Scala we have used below 3 classes namely Circle, Rectangle and Cuboid to calculate area and volume of cuboid-

Below screenshot is for class Circle which takes 1 argument as input and is defining a method named area which calculates the area of the Circle-

```
⊕ ÷ | ‡ · l<sup>+</sup>
Project
                                                 Circle.scala ×
                                                                 Rectangle.scala
                                                                                     Cuboi 🚰
  Assignment-15.1 [assignment-15-1] C:\Users\rakes\ld 1
                                                   class Circle(val 1: Int) [
  > idea
                                                          def area(x: Int)
    project [assignment-15-1-build] sources root
                                                            val a: Int = 22 / 7 * x * x
                                                4
    src
                                                            println {
                                                5
     main
                                                              "Area of circle is" + a
                                                6
        scala
                                                7
              Area
                                                8
                                                9
              Circle
              Cuboid 🖳
              🥵 Rectangle
```

Below screenshot is for class Rectangle which is extending the class Circle and also calculating area of rectangle based on different definition. This takes 2 integers as input-

```
⊕ ÷ | ÷ I<sup>*</sup>
                                              Circle.scala X
                                                              Rectangle.scala × Cuboid.scala ×
Assignment-15.1 [assignment-15-1] C:\Users\rakes\ld 1
                                                @_ class Rectangle(override val 1:Int, val b:Int ) extends Circle(1) {
                                                       def area(x: Int, y: Int) {
> project [assignment-15-1-build] sources root
                                                         val a: Int = x * y
∨ 🖿 src
                                                         println("Area of rectangle is" + a)
   main
                                             6
      scala
            O Area
            Circle
            Cuboid 🚅
           Rectangle
   > test
```

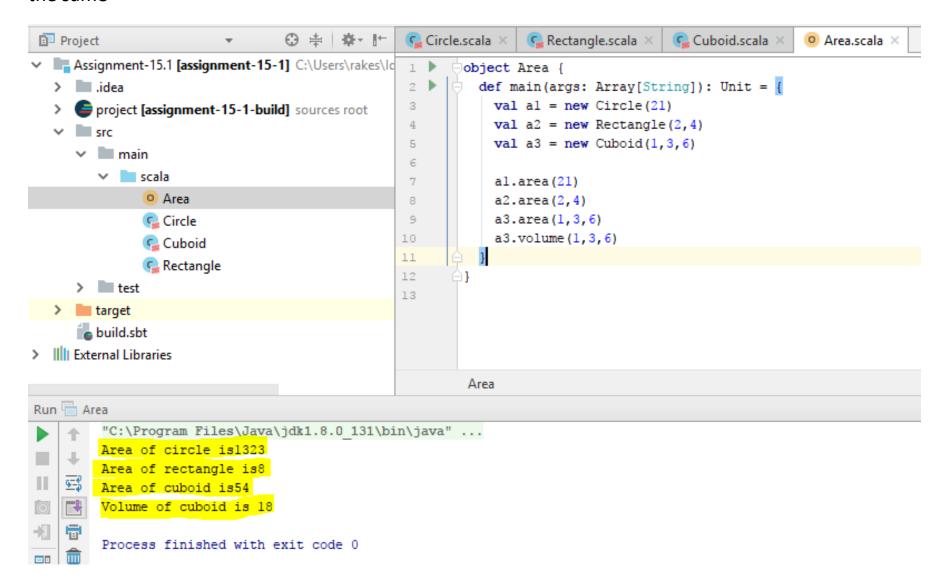
Below screenshot shows the class definition of Cuboid which takes 3 variables as input and is extending Rectangle class. It contains 2 methods area which calculates surface area of Cuboid and volume which calculates volume of cuboid-

```
⊕ + + ⊩
                                               Circle.scala × | GRectangle.scala ×
                                                                                  Cuboid.scala ×
                                                                                                   O Area.scala ×
Project
Assignment-15.1 [assignment-15-1] C:\Users\rakes\ld
                                                      class Cuboid (override val 1:Int, override val b:Int, val h:Int) extends Rectangle(1,b)
                                                        def area (x:Int, y:Int, z:Int) {
  > e project [assignment-15-1-build] sources root
                                                           val a = 2*(x*y + y*z + x*z)
                                                4
  ∨ III src
                                                           println("Area of cuboid is" + a)
                                                5

✓ Imain

                                                6
        scala
              Area
                                                      def volume (x:Int, y:Int, z:Int) {
                                               8
                                                          val v = x * y * z
              Circle
                                                9
                                                          println("Volume of cuboid is " + v)
              Cuboid 🚅
                                               11
              🖳 Rectangle
                                               12
                                                   ₽}
     > test
```

Below screenshot shows the definition of main object Area in which we are passing values to the object of the above classes and calculating the area and volume. Bottom part shows the results for the same-



## **Problem Statement**

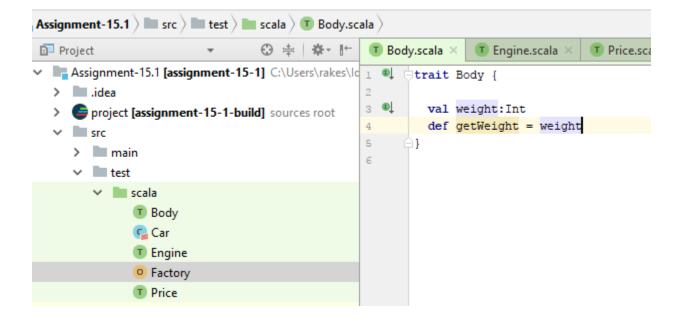
2. Write a simple program to show multiple inheritances in scala.

### **Solution**

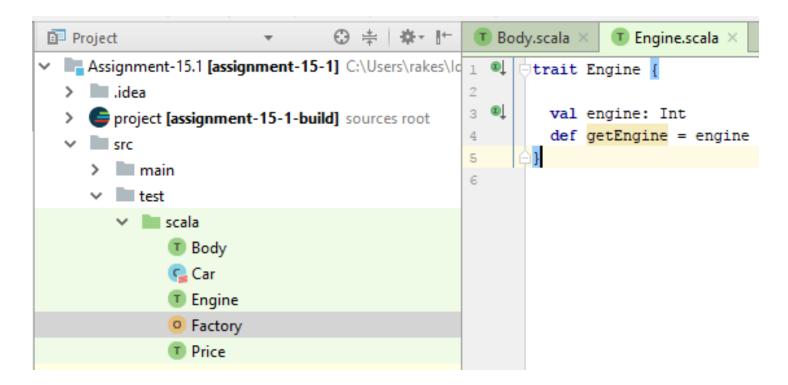
In Scala multiple inheritances is implemented via <u>traits</u>. A trait encapsulates method and field definitions, which can then be reused by mixing them into classes. Unlike class inheritance, in which each class must inherit from just one superclass, a class can mix in any number of traits.

Here we are using below Traits to show multiple inheritance in Scala

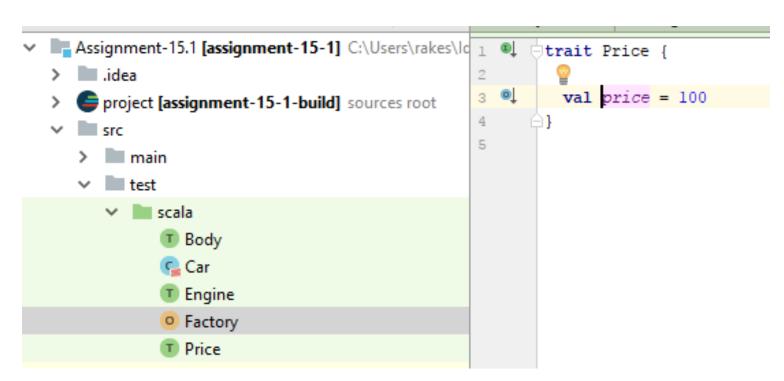
Below is the Trait written named as body which define one val named as weight and one function named as getWeight.



Below is the Trait written named as Engine which define one val named as engine and one function named as getEngine.



Below is the trait named as Price where we are assigning a val price as 100



Below is the class written which is inheriting all the properties and parameters defined in above traits. It is extending trait Body with Engine with trait Price

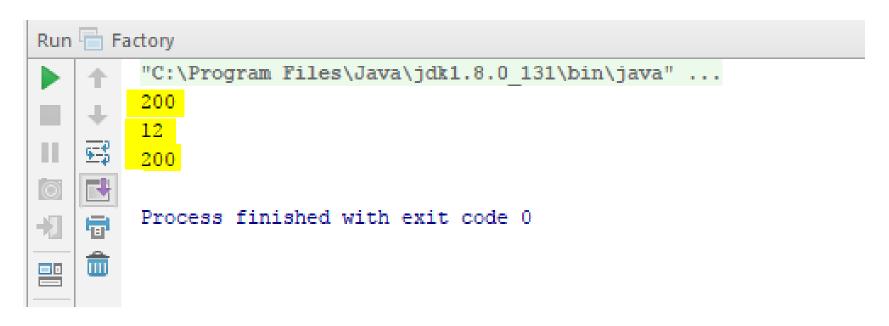


Below is the object defined to print values of all vals and methods defined in above class. Here we are instantiating object of Car and calling the different methods defined in separate traits-

```
    Assignment-15.1 [assignment-15-1] C:\Users\rakes\ld

                                                    object Factory {
                                                    def main(args: Array[String]): Unit = {
  project [assignment-15-1-build] sources root
                                                         val car: Car = new Car
                                                          println(car.weight)
     > main
                                                         println(car.engine)
     test
                                                         println(car.price)
        scala
                                               8
                                                     ₽ }
                                                    ♠}
                                               9
              Body
                                              10
              Car
              Engine
              Factory
              Price
```

Below is the final screenshot of output-



#### **Problem Statement**

1. 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-**

We have defined below class **PartialFunc** which contains partial function which takes 2 integers from user and 1 constant as input and adds the result.

We have 1 more method in same class named as square which takes the above function and squares the result-

```
RartialFunc.scala X
   Project
                                      ⊕ + + - 1-
                                                                              PartialApply.scala ×

    Assignment-15.2 [assignment-15-2] C:\Users\rakes\ld 1

                                                               class PartialFunc {
                                                                  val addConstantTo: PartialFunction[(Int, Int), Int] = {
         e project [assignment-15-2-build] sources root
                                                                    case(a,b) => a+b+10
                                                        4

♣ 
☐: Structure

                                                        5
            main
                                                        6
            scala
                                                                  def square(addConstantTo:Int) = addConstantTo * addConstantTo
                                                        7
                  PartialApply
                                                        8
                  强 PartialFunc
                                                        9
```

Now in the main method we are calling same function. We are passing 10, 20 as input and constant 10 is being added.

```
⊕ \ \dot\ \dot
     Project
                                                                                                                                                                                                                                                                                                                                                            🗲 PartialFunc.scala 🗵
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           PartialApply.scala ×

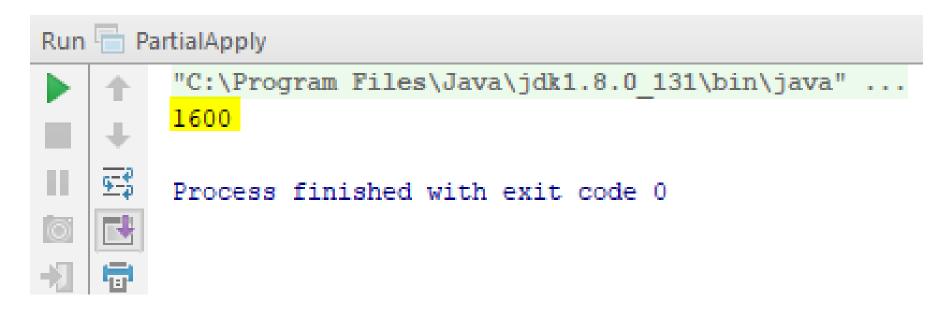
    Assignment-15.2 [assignment-15-2] C:\Users\rakes\ld

                                                                                                                                                                                                                                                                                                                                                                                                              object PartialApply {
                                                                                                                                                                                                                                                                                                                                                             3
                                                                                                                                                                                                                                                                                                                                                                                                                           def main(args: Array[String]): Unit = {
                                     project [assignment-15-2-build] sources root
                                                                                                                                                                                                                                                                                                                                                                                                                                            val x = new PartialFunc
                                         main
                                                                                                                                                                                                                                                                                                                                                                                                                                          println(x.square(x.addConstantTo((10,20))))
                                                             scala
                                                                                                                                                                                                                                                                                                                                                             7

    PartialApply

                                                                                                                                                                                                                                                                                                                                                             8
                                                                                                                                                                                                                                                                                                                                                             9
                                                                                                         PartialFunc
```

Below is the output-



# **Problem Statement**

1. Write a program to print the prices of 4 courses of Acadgild: Android-12999, Big Data Development-17999, Big Data Development-17999, Spark-19999 using match and add a default condition if the user enters any other course.

#### **Solution-**

We have defined a class which contains a method for pattern matching which will take string as input which will be course name in our case and will give its price(Int) as output-

Below is the main method in which we are calling this method using course name and its corresponding price is getting generated-

Below is the output (course price) for the same-