

Session 17: SCALA BASICS

4 Assignment 1

Task 1

Write a simple program to show inheritance in scala.

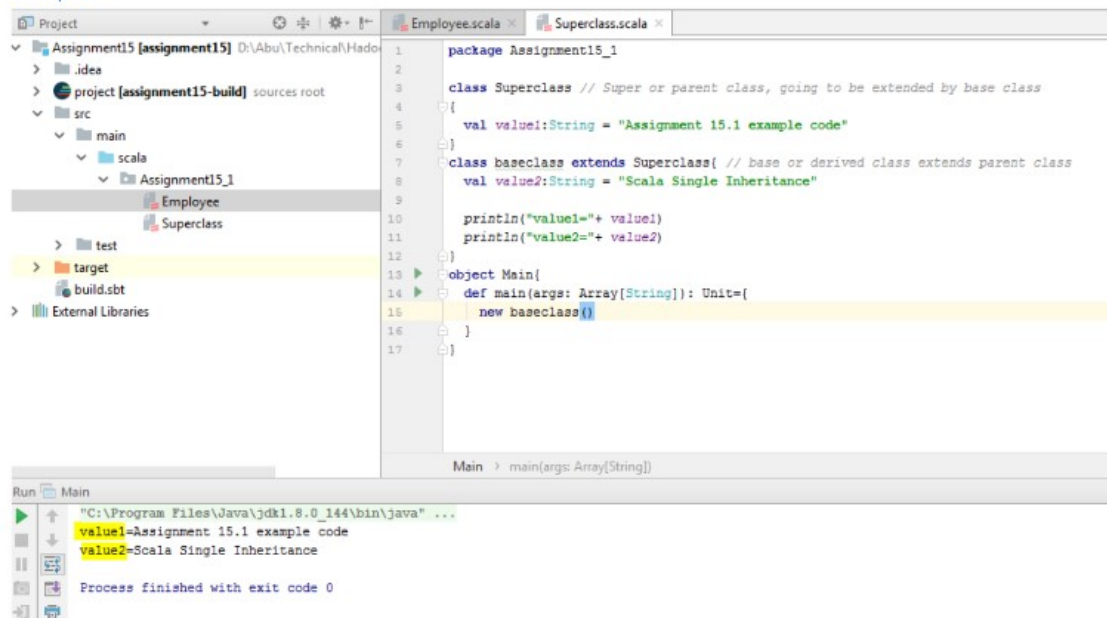
Scala Code

```
package Assignment15_1

class Superclass // Super or parent class, going to be extended by base class
{
    val value1:String = "Assignment 15.1 example code"
}
class baseclass extends Superclass{ // base or derived class extends parent class
    val value2:String = "Scala Single Inheritance"

    println("value1="+ value1)
    println("value2="+ value2)
}
object Main{
    def main(args: Array[String]): Unit={
        new baseclass()
    }
}
```

Output



Task 2

Write a simple program to show multiple inheritance in scala

```
package Assignment15_1

trait MultipleInheritance //parent trait
{
  def show() // defining the function show()
  {
    println("Assignment 15.1")
  }
}

trait one extends MultipleInheritance // extending the parent trait
{
  override def show()
  {
    println("This won't be printed")
  }
}

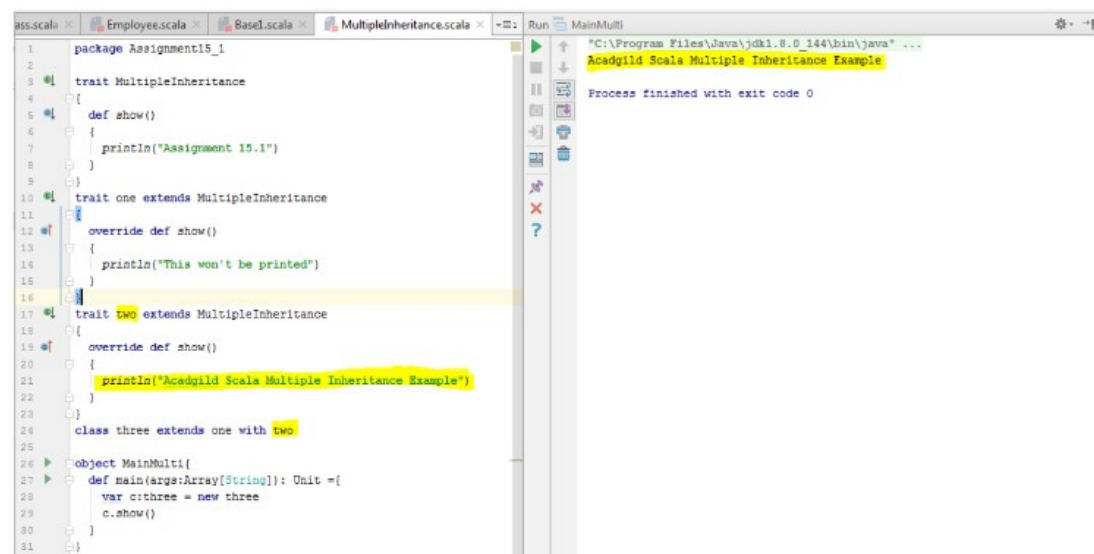
trait two extends MultipleInheritance // extending the parent trait
{
  override def show()
  {
    println("Acadgild Scala Multiple Inheritance Example")
  }
}

class three extends one with two //extending the base traits, calling the function show()

object MainMulti{
  def main(args:Array[String]): Unit ={
    var c:three = new three // it will call last function which is mentioned in the class three, changing the order will give different result
    c.show()
  }
}
```

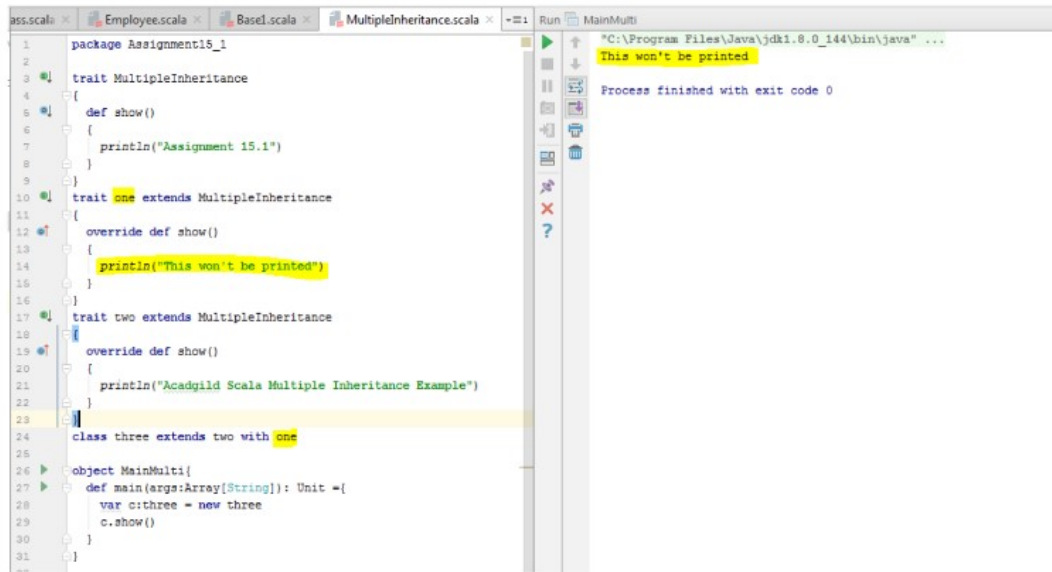
Output

Example 1, here the class **three** calling the trait one with **two**, the **two** in the last order and hence the function of **two** will be called and output is,



The screenshot shows an IDE with a Scala file named `MultipleInheritance.scala` and a `Run` window. The code in the editor is identical to the one in the previous block. The `Run` window shows the command `"C:\Program Files\Java\jdk1.8.0_144\bin\java" ...` and the output `Acadgild Scala Multiple Inheritance Example`. Below the output, it says "Process finished with exit code 0".

Example 2, in this example the object *MainMulti* called the trait *one* and see the result below,



The screenshot shows an IDE with a Scala file named `MultipleInheritance.scala` and a Run console. The code defines a package `Assignment15_1` with a trait `MultipleInheritance` containing a `show()` method that prints "Assignment 15.1". Two other traits, `one` and `two`, extend `MultipleInheritance`. `one` overrides `show()` to print "This won't be printed". `two` overrides `show()` to print "Acadgild Scala Multiple Inheritance Example". A class `three` extends `two` and mixes in `one`. An object `MainMulti` contains a `main` method that creates an instance of `three` and calls `show()`. The Run console shows the output of the `main` method, which is "This won't be printed", indicating that the `one` trait's `show()` method was called.

```
1 package Assignment15_1
2
3 trait MultipleInheritance
4 {
5   def show()
6   {
7     println("Assignment 15.1")
8   }
9 }
10 trait one extends MultipleInheritance
11 {
12   override def show()
13   {
14     println("This won't be printed")
15   }
16 }
17 trait two extends MultipleInheritance
18 {
19   override def show()
20   {
21     println("Acadgild Scala Multiple Inheritance Example")
22   }
23 }
24 class three extends two with one
25
26 object MainMulti{
27   def main(args:Array[String]): Unit ={
28     var c:three = new three
29     c.show()
30   }
31 }
```

Run MainMulti

"C:\Program Files\Java\jdk1.8.0_144\bin\java" ...
This won't be printed
Process finished with exit code 0