

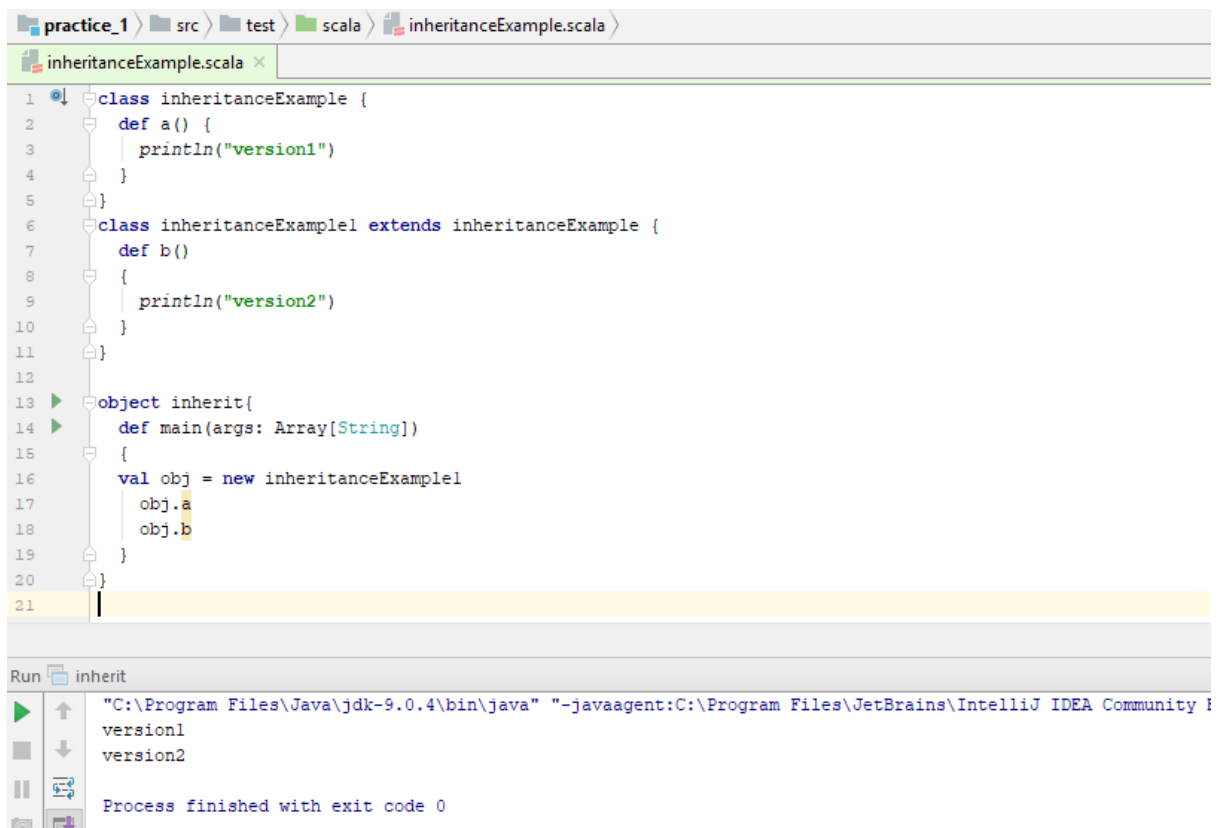
Assignment17.1

Task 1

Write a simple program to show inheritance in Scala.

In the below screen shot **inheritanceExample** class is extends by **inheritanceExample1** class. We are able to obtain the output of method **a()** and **b()** in object **inherit**.

It prints the output **version1** and **version2** from object **inherit** in print screen.



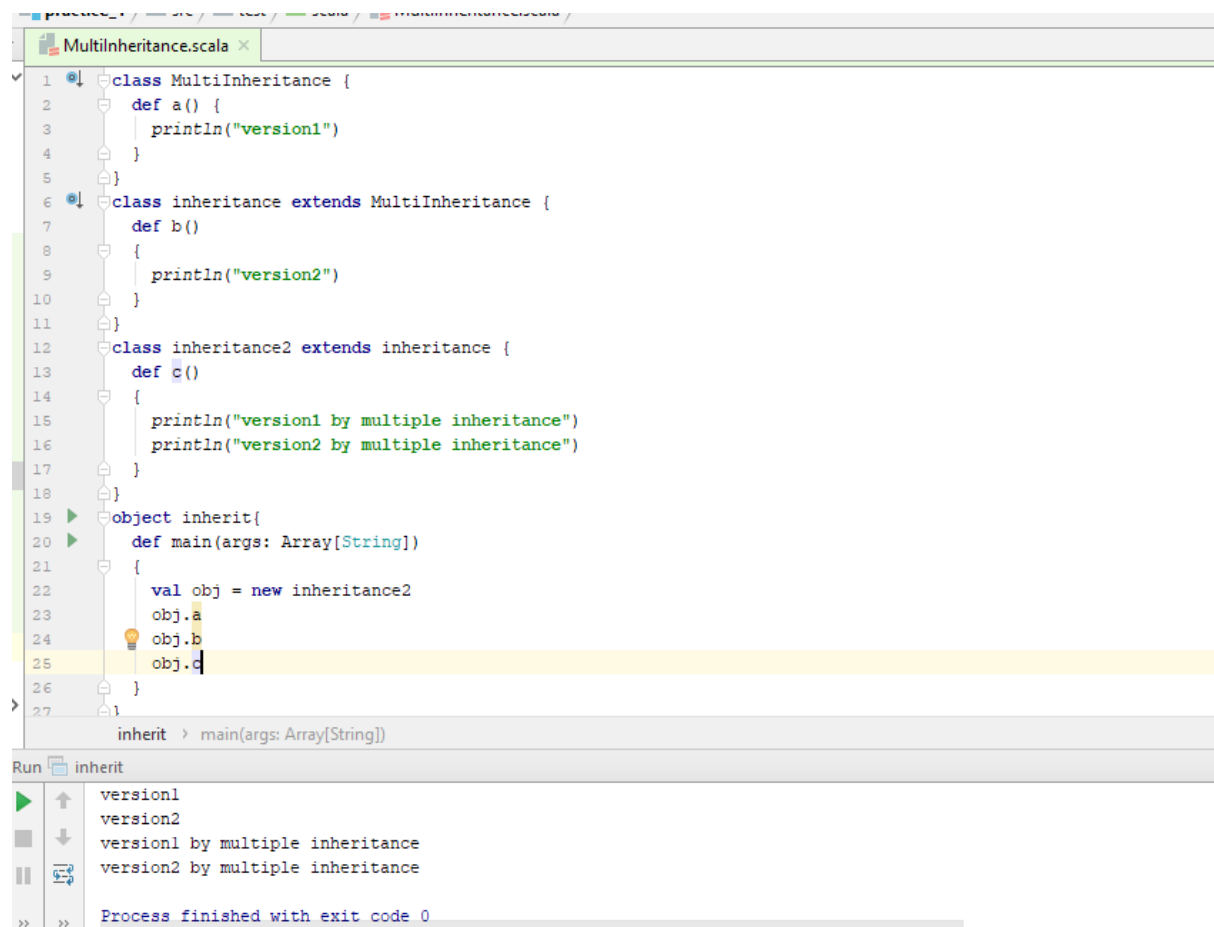
```
practice_1 > src > test > scala > inheritanceExample.scala
inheritanceExample.scala x
1 class inheritanceExample {
2   def a() {
3     println("version1")
4   }
5 }
6 class inheritanceExample1 extends inheritanceExample {
7   def b()
8   {
9     println("version2")
10  }
11 }
12
13 object inherit{
14   def main(args: Array[String])
15   {
16     val obj = new inheritanceExample1
17     obj.a
18     obj.b
19   }
20 }
21

Run inherit
"C:\Program Files\Java\jdk-9.0.4\bin\java" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community I
version1
version2
Process finished with exit code 0
```

Task 2

Write a simple program to show multiple inheritance in Scala.

In below screen shot we performed multiple inheritance, class **inheritance** extends **MultiInheritance** and inheritance2 extends to **inheritance**.



```
1 class MultiInheritance {
2   def a() {
3     println("version1")
4   }
5 }
6 class inheritance extends MultiInheritance {
7   def b() {
8     println("version2")
9   }
10 }
11 class inheritance2 extends inheritance {
12   def c() {
13     println("version1 by multiple inheritance")
14     println("version2 by multiple inheritance")
15   }
16 }
17 object inherit {
18   def main(args: Array[String]) {
19     val obj = new inheritance2
20     obj.a
21     obj.b
22     obj.c
23   }
24 }
25 inherit > main(args: Array[String])
```

Run inherit

```
version1
version2
version1 by multiple inheritance
version2 by multiple inheritance
Process finished with exit code 0
```

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.

```
Function.scala
PartialFunctionResult
PartialFunction.scala x
1 class PartialFunction {
2
3     def summation(a:Int,b:Int,c:Int) = a+b+c //here we have defined the sum function
4
5     def partialsum(x:Int,y:Int) {
6
7         val add = summation(_: Int, _: Int, 10) //here one value defined as constant by defining it in partial function
8         println("sum of the numbers:" + add(x,y))
9
10        def squareResult(result: Int) = result * result
11
12        val square = squareResult(add(x,y))
13        println("square of the resultant sum:" + square )
14    }
15 }
16
17 object PartialFunctionResult {
18
19     def main(args: Array[String]): Unit = {
20
21         println("Enter the numbers")
22
23         var x:Int = scala.io.StdIn.readInt()
24         var y:Int = scala.io.StdIn.readInt()
25
26         new PartialFunction().partialsum(x,y)
27     }
28 }
```

Output of the above code:

```
Run PartialFunctionResult
"C:\Program Files\Java\jdk-9.0.4\bin\java" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Editio
Enter the numbers
9
5
sum of the numbers:24
square of the resultant sum:576
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.

Below screen shot shows the program to print the prices for the courses:



```
1  object CoursesProgram {
2
3      def courseMatch(x: String): String = x match
4
5      {
6          case "Android App Development" => "price is 14,999 INR"
7          case "Data Science" => " price is 49,999 INR"
8          case "Big Data Hadoop & Spark Developer" => " price is 24,999 INR"
9          case "Blockchain Certification" => "price is 49,999 INR"
10         case _ => "Sorry,this course is not available"
11     }
12
13     def main(args: Array[String])
14     {
15         println(courseMatch("Data Science"))
16         println(courseMatch("Python"))
17     }
18 }
19
```

CoursesProgram

"C:\Program Files\Java\jdk-9.0.4\bin\java" "-javaagent:C:\Program Files\JetBrains\Intel
price is 49,999 INR
Sorry,this course is not available
Process finished with exit code 0

acadgild@localhost:~

```
scala> def courseMatch(x: String): String = x match
|
|   {
|     case "Android App Development" => "price is 14,999 INR"
|     case "Data Science" => " price is 49,999 INR"
|     case "Big Data Hadoop & Spark Developer" => " price is 24,999 INR"
|     case "Blockchain Certification" => "price is 49,999 INR"
|     case _ => "Sorry,this course is not available"
|   }
courseMatch: (x: String)String

scala> courseMatch("Big Data Hadoop & Spark Developer")
res6: String = " price is 24,999 INR"

scala> courseMatch("Java")
res7: String = Sorry,this course is not available
```