

## Session 17: SCALA BASICS 4

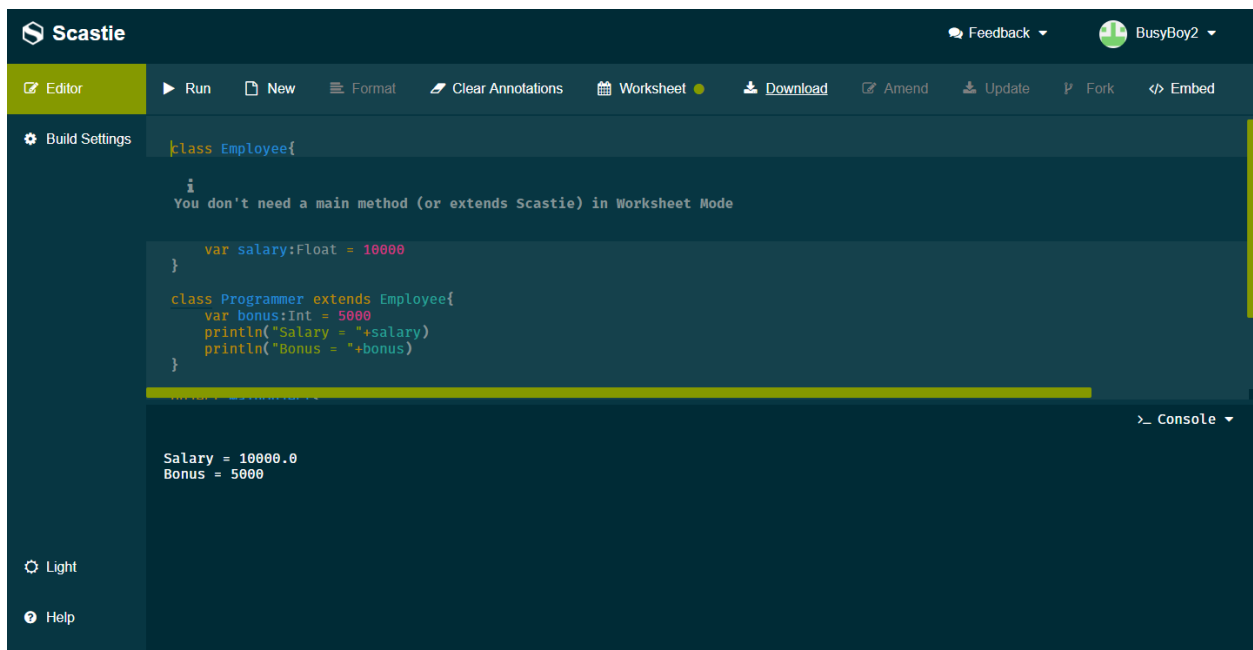
### Task 1

Write a simple program to show inheritance in Scala.

```
class Employee{
  var salary:Float = 10000
}

class Programmer extends Employee{
  var bonus:Int = 5000
  println("Salary = "+salary)
  println("Bonus = "+bonus)
}

object MainObject{
  def main(args:Array[String]){
    new Programmer()
  }
}
```



The screenshot shows the Scastie web interface for running Scala code. The top bar includes the Scastie logo, a feedback button, and a user profile for 'BusyBoy2'. Below the top bar is a navigation menu with options: Editor, Run, New, Format, Clear Annotations, Worksheet, Download, Amend, Update, Fork, and Embed. The main editor area displays the Scala code from the previous block. A message indicates that a main method is not required in Worksheet Mode. The console output at the bottom shows the results of running the code: 'Salary = 10000.0' and 'Bonus = 5000'. The interface also includes a 'Build Settings' sidebar on the left and a 'Console' dropdown at the bottom right.

```
class Employee{
  var salary:Float = 10000
}

class Programmer extends Employee{
  var bonus:Int = 5000
  println("Salary = "+salary)
  println("Bonus = "+bonus)
}

object MainObject{
  def main(args:Array[String]){
    new Programmer()
  }
}
```

You don't need a main method (or extends Scastie) in Worksheet Mode

Salary = 10000.0  
Bonus = 5000

## Session 17: SCALA BASICS 4

### Task 2

Write a simple program to show multiple inheritance in Scala

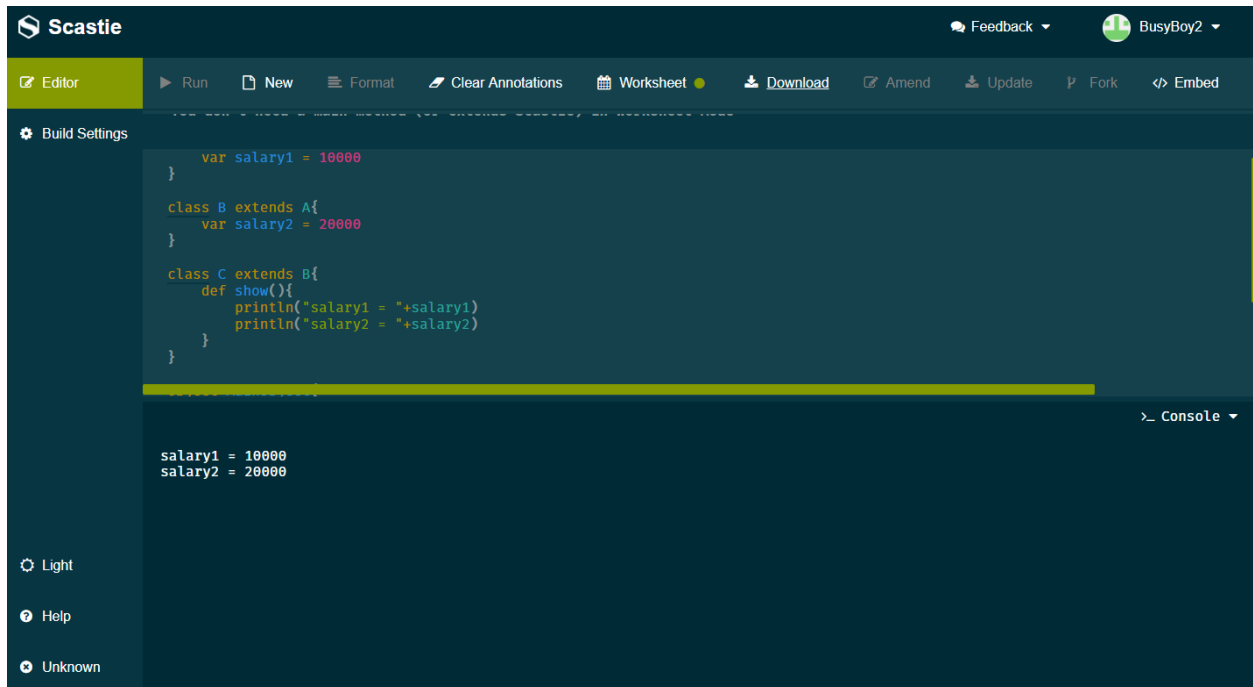
```
class A{
  var salary1 = 10000
}

class B extends A{
  var salary2 = 20000
}

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

object MainObject{

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



The screenshot shows the Scastie online Scala IDE interface. The top bar includes the Scastie logo, a feedback button, and a user profile for 'BusyBoy2'. Below the top bar is a navigation menu with options: Editor (selected), Run, New, Format, Clear Annotations, Worksheet, Download, Amend, Update, Fork, and Embed. The main editor area displays the Scala code for multiple inheritance. On the left, there is a 'Build Settings' sidebar. At the bottom, a 'Console' panel shows the output of the program: 'salary1 = 10000' and 'salary2 = 20000'.

```
var salary1 = 10000
}

class B extends A{
  var salary2 = 20000
}

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

object MainObject{

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

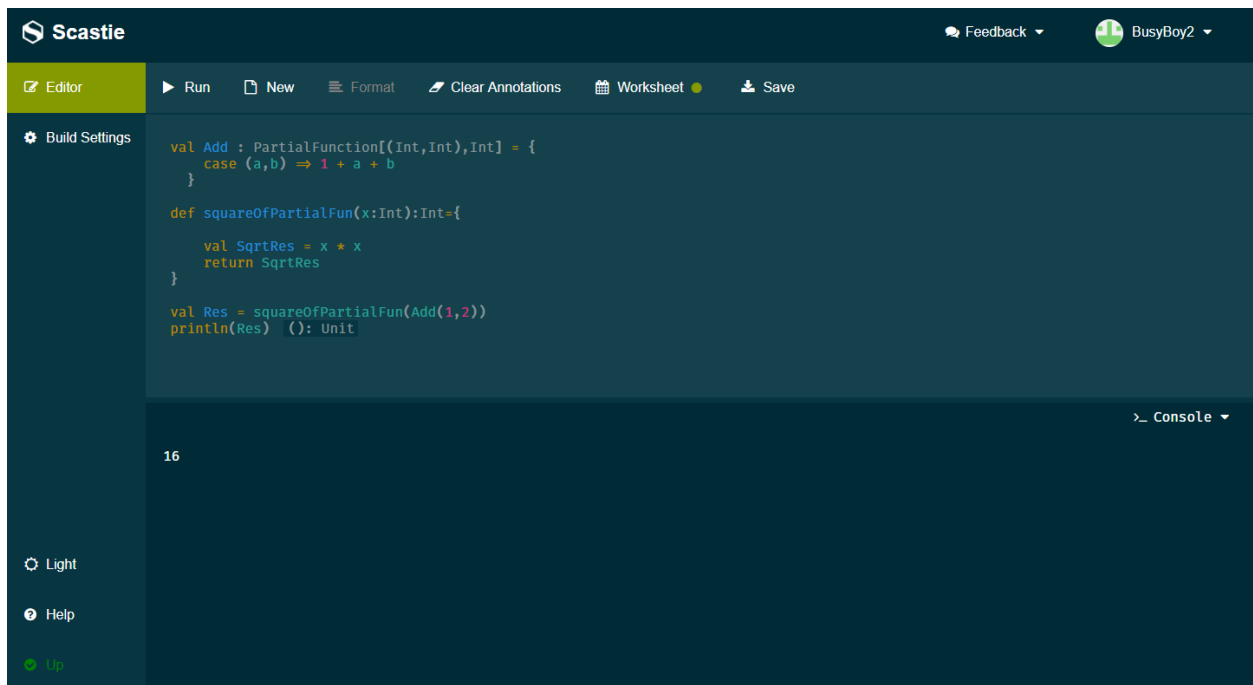
salary1 = 10000  
salary2 = 20000

## Session 17: SCALA BASICS 4

### 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.

```
val Add : PartialFunction[(Int,Int),Int] = {  
  case (a,b) => 1 + a + b  
}  
  
def squareOfPartialFun(x:Int):Int={  
  val SqrtRes = x * x  
  return SqrtRes  
}  
  
val Res = squareOfPartialFun(Add(1,2))  
println(Res)
```



## Session 17: SCALA BASICS 4

### 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.

```
class CaseClass{  
  def Matching(x: String): String = x match {  
    case "Android App Development"      => "14,999 INR"  
    case "Data Science"                 => "49,999 INR"  
    case "Big Data Hadoop & Spark Developer" => "24,999 INR"  
    case "Blockchain Certification"      => "49,999 INR"  
    case _                              => "Please enter relavent course name"  
  }  
}  
  
object CaseMatching extends App{  
  val Match = new CaseClass()  
  
  println("Course Name: " + "Android App Development" + "\nCourse Fee: "+Match.Matching("Android App Development"))  
  println("Course Name: " + "Data Science" + "\nCourse Fee: "+Match.Matching("Data Science"))  
  println("Course Name: " + "Big Data Hadoop & Spark Developer" + "\nCourse Fee: "+Match.Matching("Big Data Hadoop & Spark Developer"))  
  println("Course Name: " + "Blockchain Certification" + "\nCourse Fee: "+Match.Matching("Blockchain Certification"))  
  println("Course Name: " + "Java Concepts" + "\nCourse Fee: "+Match.Matching("Java Concepts"))  
}
```

The screenshot shows the Scastie online Scala IDE interface. At the top, there's a header with the Scastie logo, a 'Feedback' button, and a user profile 'BusyBoy2'. Below the header is a toolbar with buttons for 'Editor', 'Run', 'New', 'Format', 'Clear Annotations', 'Worksheet', 'Download', 'Amend', 'Update', 'Fork', and 'Embed'. The main editor area displays the Scala code from the previous block. A message below the code states: 'You don't need a main method (or extends Scastie) in Worksheet Mode'. At the bottom, the 'Console' tab is active, showing the output of the program: 'Course Name: Android App Development', 'Course Fee: 14,999 INR', 'Course Name: Data Science', 'Course Fee: 49,999 INR', 'Course Name: Big Data Hadoop & Spark Developer', 'Course Fee: 24,999 INR', 'Course Name: Blockchain Certification', 'Course Fee: 49,999 INR', 'Course Name: Java Concepts', and 'Course Fee: Please enter relavent course name'.