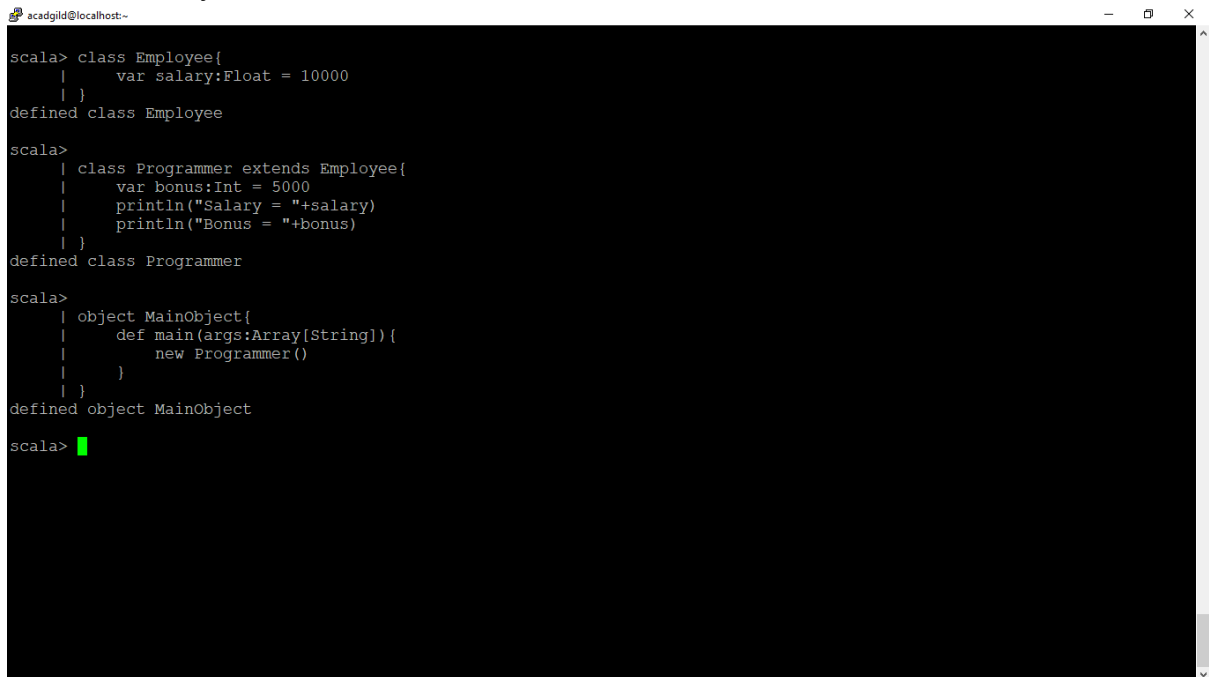


## ASSIGNMENT

**Task1:** Write a simple program to show inheritance in scala.

**Sol:**

```
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()
    }
}
```

A screenshot of a Scala REPL window titled 'acadgild@localhost:~'. The window shows the following code being entered and executed:

```
scala> class Employee{
|   var salary:Float = 10000
| }
defined class Employee

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

scala> object MainObject{
|   def main(args:Array[String]){
|       new Programmer()
|   }
| }
defined object MainObject

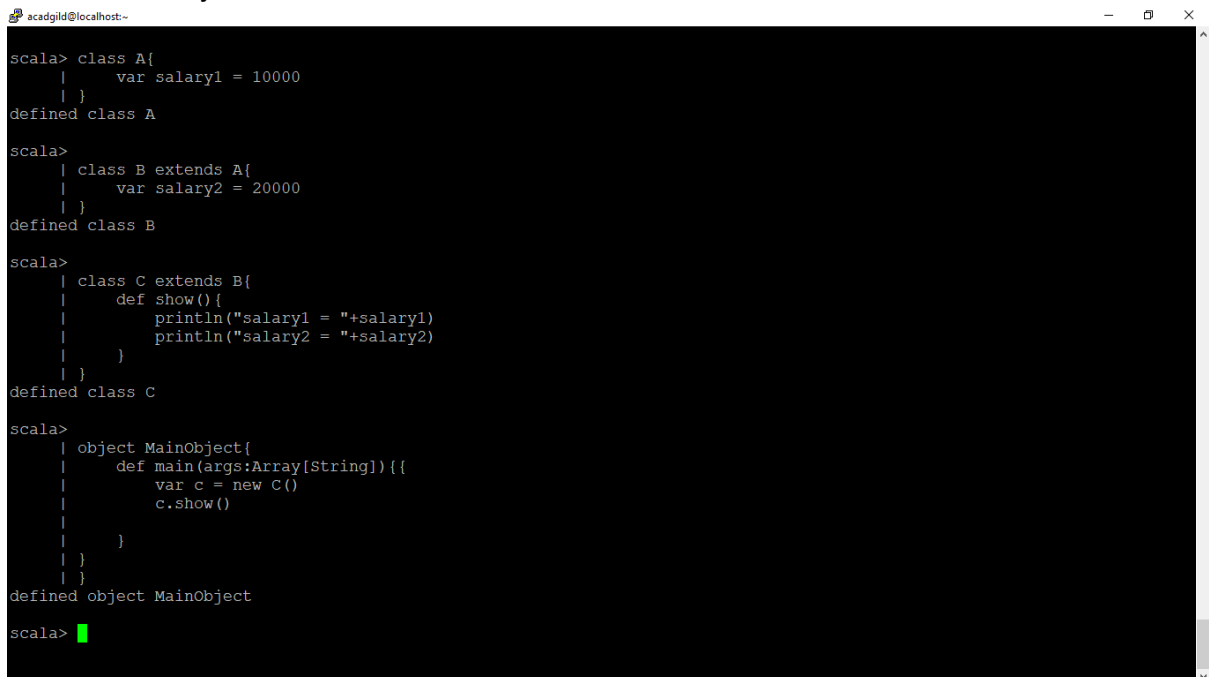
scala>
```

The output shows the successful compilation of the classes and the object, with no runtime errors or output from the program itself.

**Task2:** Write a simple program to show multiple inheritance in scala

**Sol:**

```
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()
    }
}
```



```
scala> class A{
|   var salary1 = 10000
| }
defined class A

scala>
| class B extends A{
|   var salary2 = 20000
| }
defined class B

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

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

scala> █
```

### Task3

1. A. Write a partial function to add two inputs and one constant value

acadgild@localhost:~

```
scala> val SumFunc = (x: Int, y: Int) => x+y+30
SumFunc: (Int, Int) => Int = $$Lambda$1054/1134505351@721bf7ad

scala> SumFunc(10,25)
res3: Int = 65

scala> SumFunc(10,100)
res4: Int = 140

scala> []
```

1. B. Define another function which takes partial function as input and prints the square

acadgild@localhost:~

```
scala> val sqFunc = (SumFunc: Int) => SumFunc*SumFunc
sqFunc: Int => Int = $$Lambda$1064/519425508@359fa722

scala> sqFunc(SumFunc(10,20))
res6: Int = 3600

scala> sqFunc(SumFunc(5,6))
res7: Int = 1681

scala> █
```

### Task4 Write a program to print the prices of 4 courses of Acadgild

**Sol:**

```
def CourseFee(x: Int): String = x match {
  case 1 => "Android App Development -14,999 INR"
  case 2 => "Data Science - 49,999 INR"
  case 3 => "Big Data Hadoop & Spark Developer – 24,999 INR"
  case 4 => "Blockchain Certification – 49,999 INR"
  case _ => "oops!! course not available" // the default, catch-all
}
```

```
scala> def CourseFee(x: Int): String = x match {  
  | case 1 => "Android App Development -14,999 INR"  
  | case 2 => "Data Science - 49,999 INR"  
  | case 3 => "Big Data Hadoop & Spark Developer - 24,999 INR"  
  | case 4 => "Blockchain Certification - 49,999 INR"  
  | case _ => "oops!! course not available" // the default, catch-all  
  | }  
CourseFee: (x: Int)String  
  
scala> CourseFee(2)  
res6: String = Data Science - 49,999 INR  
  
scala> CourseFee(4)  
res7: String = Blockchain Certification - 49,999 INR  
  
scala> CourseFee(8)  
res8: String = oops!! course not available  
  
scala> CourseFee(1)  
res9: String = Android App Development -14,999 INR  
  
scala> █
```