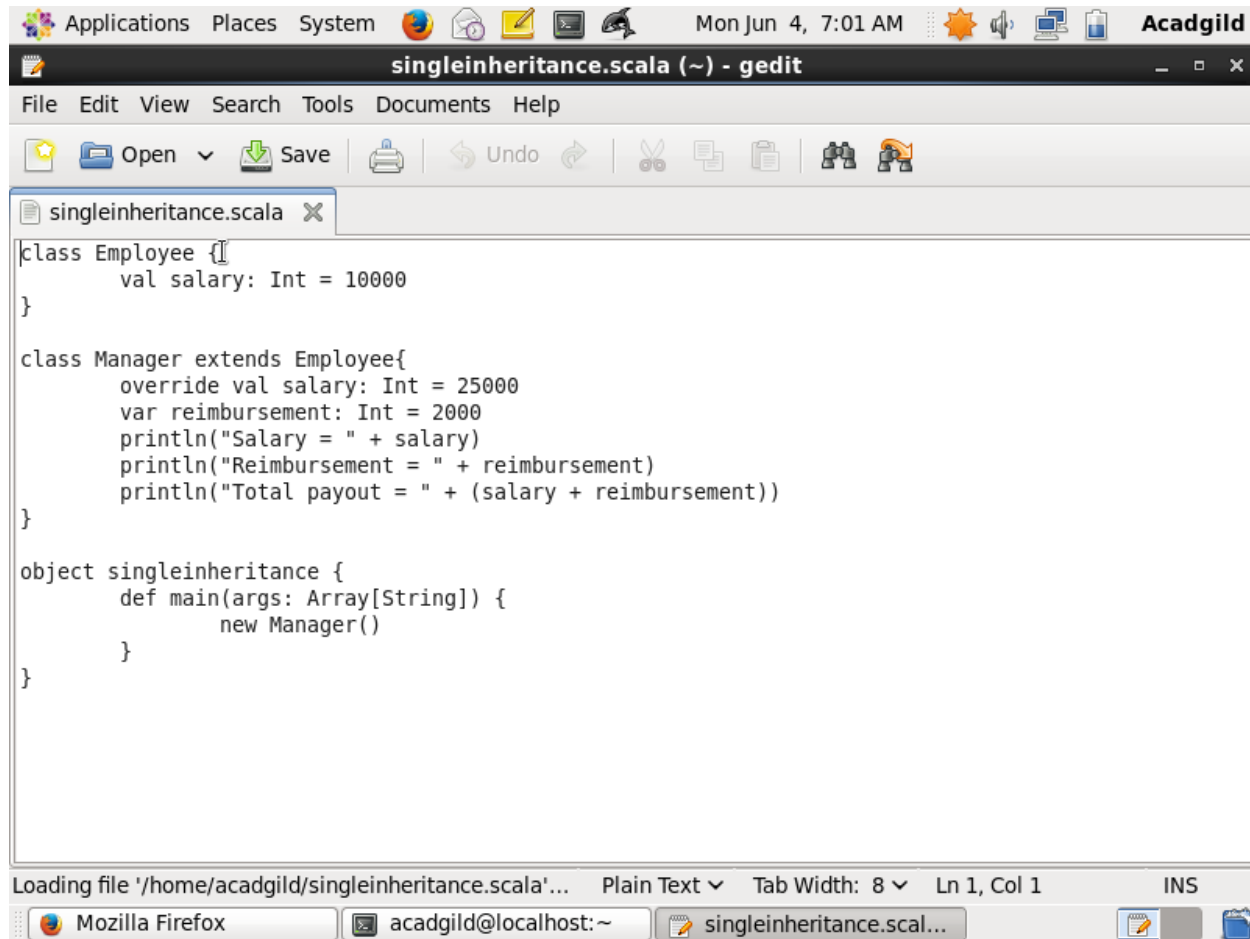


Task 1:

Employee is the super class and since Manager class extends, it becomes a sub class.

Code:

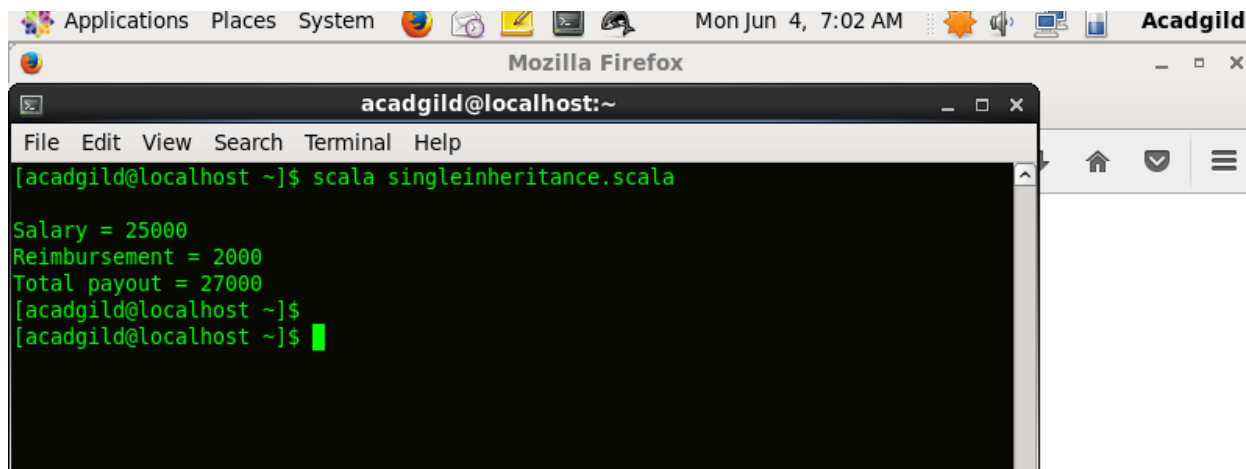


The screenshot shows a gedit editor window titled 'singleinheritance.scala (~) - gedit'. The code defines an 'Employee' class with a 'salary' attribute, a 'Manager' class that extends 'Employee' and overrides 'salary' while adding a 'reimbursement' attribute and a 'main' method, and an object 'singleinheritance' with a 'main' method that creates a 'Manager' instance.

```
class Employee {  
    val salary: Int = 10000  
}  
  
class Manager extends Employee{  
    override val salary: Int = 25000  
    var reimbursement: Int = 2000  
    println("Salary = " + salary)  
    println("Reimbursement = " + reimbursement)  
    println("Total payout = " + (salary + reimbursement))  
}  
  
object singleinheritance {  
    def main(args: Array[String]) {  
        new Manager()  
    }  
}
```

The status bar at the bottom indicates the file path, encoding (Plain Text), tab width (8), line and column numbers (Ln 1, Col 1), and the current cursor position (INS).

OutPut:



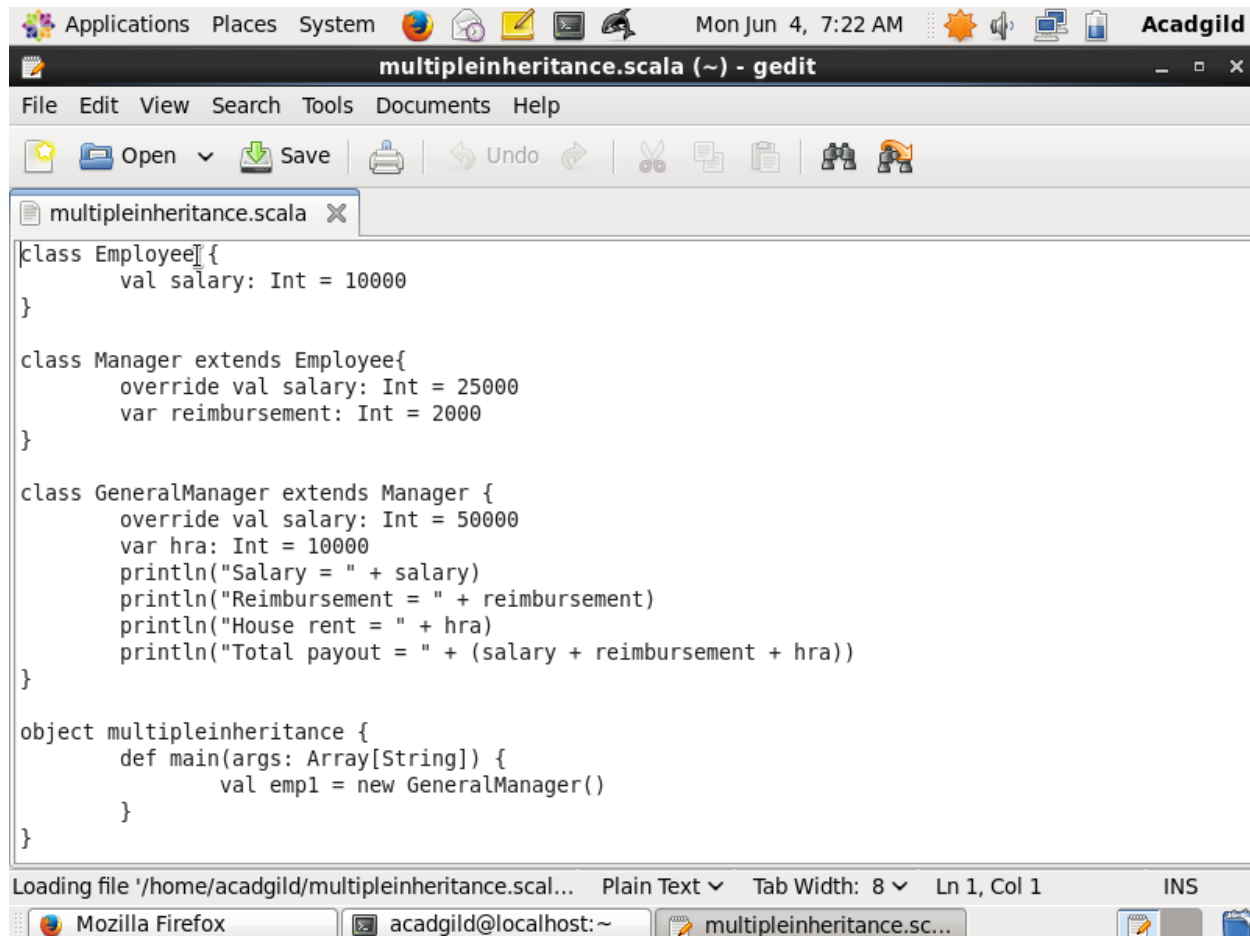
The screenshot shows a terminal window titled 'acadgild@localhost:~' with the output of the command 'scala singleinheritance.scala'. The output displays the salary, reimbursement, and total payout for a Manager object.

```
[acadgild@localhost ~]$ scala singleinheritance.scala  
  
Salary = 25000  
Reimbursement = 2000  
Total payout = 27000  
[acadgild@localhost ~]$  
[acadgild@localhost ~]$
```

Task 2 :

To show multiple inheritance I have defined one more class GeneralManager which inherits class Manager which in turn inherits Employee.

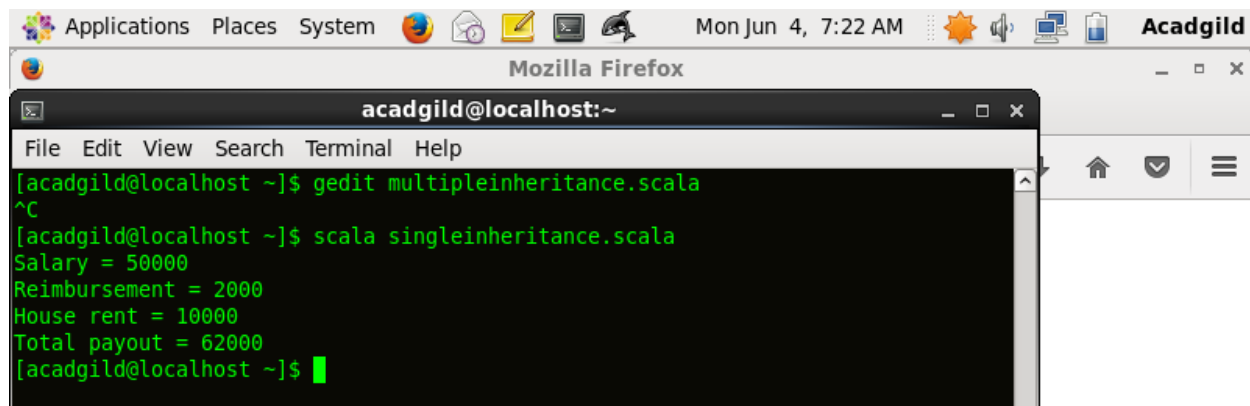
Both Manager and GeneralManager class have their own implementations for member salary and reimbursement is being inherited in subclass from another subclass showing multiple inheritance.



The screenshot shows a gedit editor window titled 'multipleinheritance.scala (~) - gedit'. The code defines three classes: Employee, Manager, and GeneralManager. Employee has a salary of 10000. Manager extends Employee, overriding salary to 25000 and adding a reimbursement of 2000. GeneralManager extends Manager, overriding salary to 50000 and adding a house rent of 10000. A main method in the multipleinheritance object prints the salary, reimbursement, house rent, and total payout for a GeneralManager instance.

```
class Employee {  
    val salary: Int = 10000  
}  
  
class Manager extends Employee {  
    override val salary: Int = 25000  
    var reimbursement: Int = 2000  
}  
  
class GeneralManager extends Manager {  
    override val salary: Int = 50000  
    var hra: Int = 10000  
    println("Salary = " + salary)  
    println("Reimbursement = " + reimbursement)  
    println("House rent = " + hra)  
    println("Total payout = " + (salary + reimbursement + hra))  
}  
  
object multipleinheritance {  
    def main(args: Array[String]) {  
        val emp1 = new GeneralManager()  
    }  
}
```

Output:



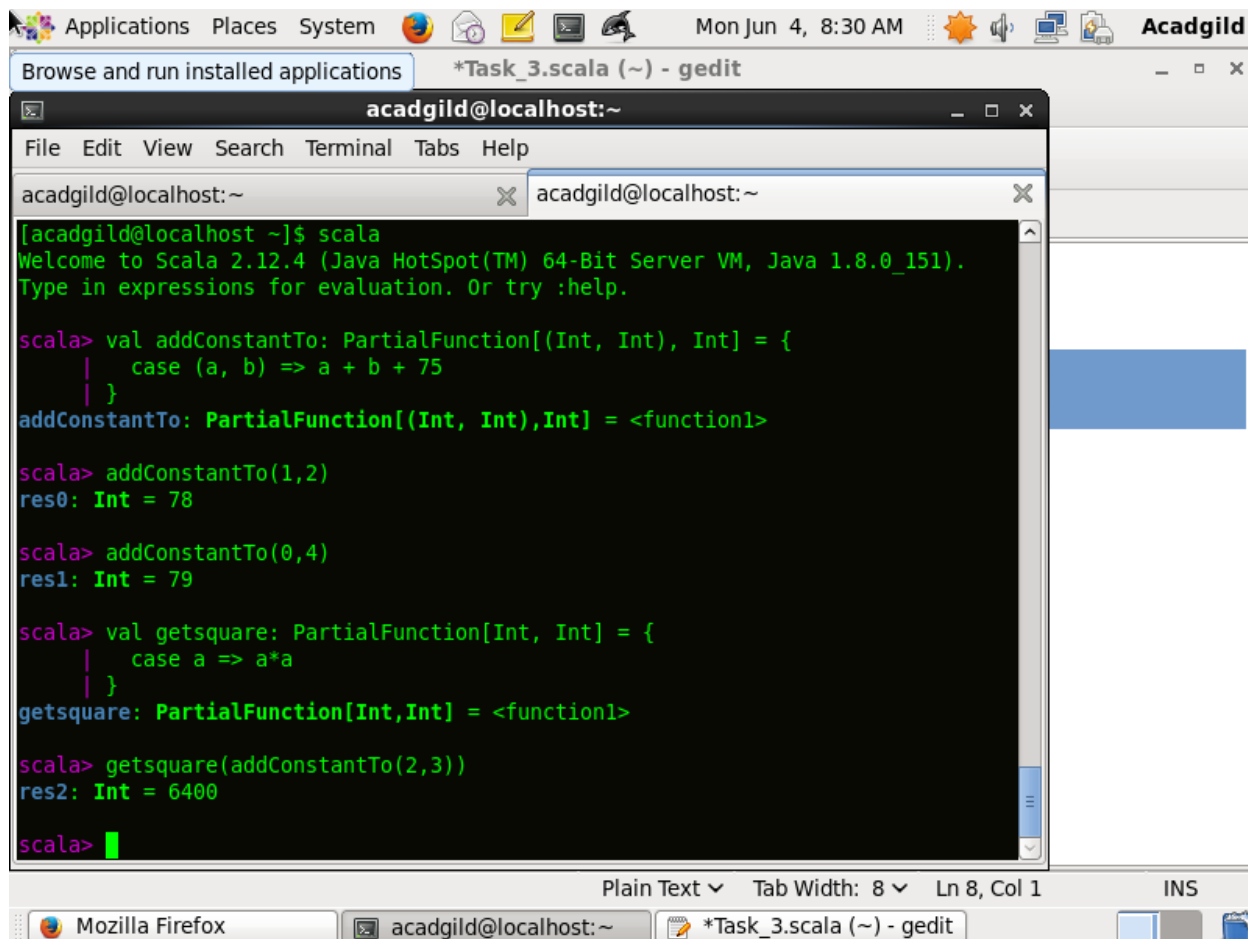
The screenshot shows a terminal window titled 'acadgild@localhost:~'. The user runs 'gedit multipleinheritance.scala' and then 'scala singleinheritance.scala'. The output shows the salary, reimbursement, house rent, and total payout for a GeneralManager instance.

```
[acadgild@localhost ~]$ gedit multipleinheritance.scala  
^C  
[acadgild@localhost ~]$ scala singleinheritance.scala  
Salary = 50000  
Reimbursement = 2000  
House rent = 10000  
Total payout = 62000  
[acadgild@localhost ~]$
```

Task 3 :

In addConstanTo partial function the first parameter is passed as tuple of two integers to accept two numbers as input to a sum of the two input numbers and a constant, 75 in this case.

another partial function gives the square so addConstantTo is passed as a parameter to getsquare



The screenshot shows a Linux desktop environment. At the top, there is a panel with icons for Applications, Places, System, and a clock showing 'Mon Jun 4, 8:30 AM'. Below this is a taskbar with several icons. The main window is a terminal titled 'acadgild@localhost:~' with a menu bar (File, Edit, View, Search, Terminal, Tabs, Help). The terminal output shows the following Scala code and results:

```
[acadgild@localhost ~]$ scala
Welcome to Scala 2.12.4 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_151).
Type in expressions for evaluation. Or try :help.

scala> val addConstantTo: PartialFunction[(Int, Int), Int] = {
  |   case (a, b) => a + b + 75
  | }
addConstantTo: PartialFunction[(Int, Int),Int] = <function1>

scala> addConstantTo(1,2)
res0: Int = 78

scala> addConstantTo(0,4)
res1: Int = 79

scala> val getsquare: PartialFunction[Int, Int] = {
  |   case a => a*a
  | }
getsquare: PartialFunction[Int,Int] = <function1>

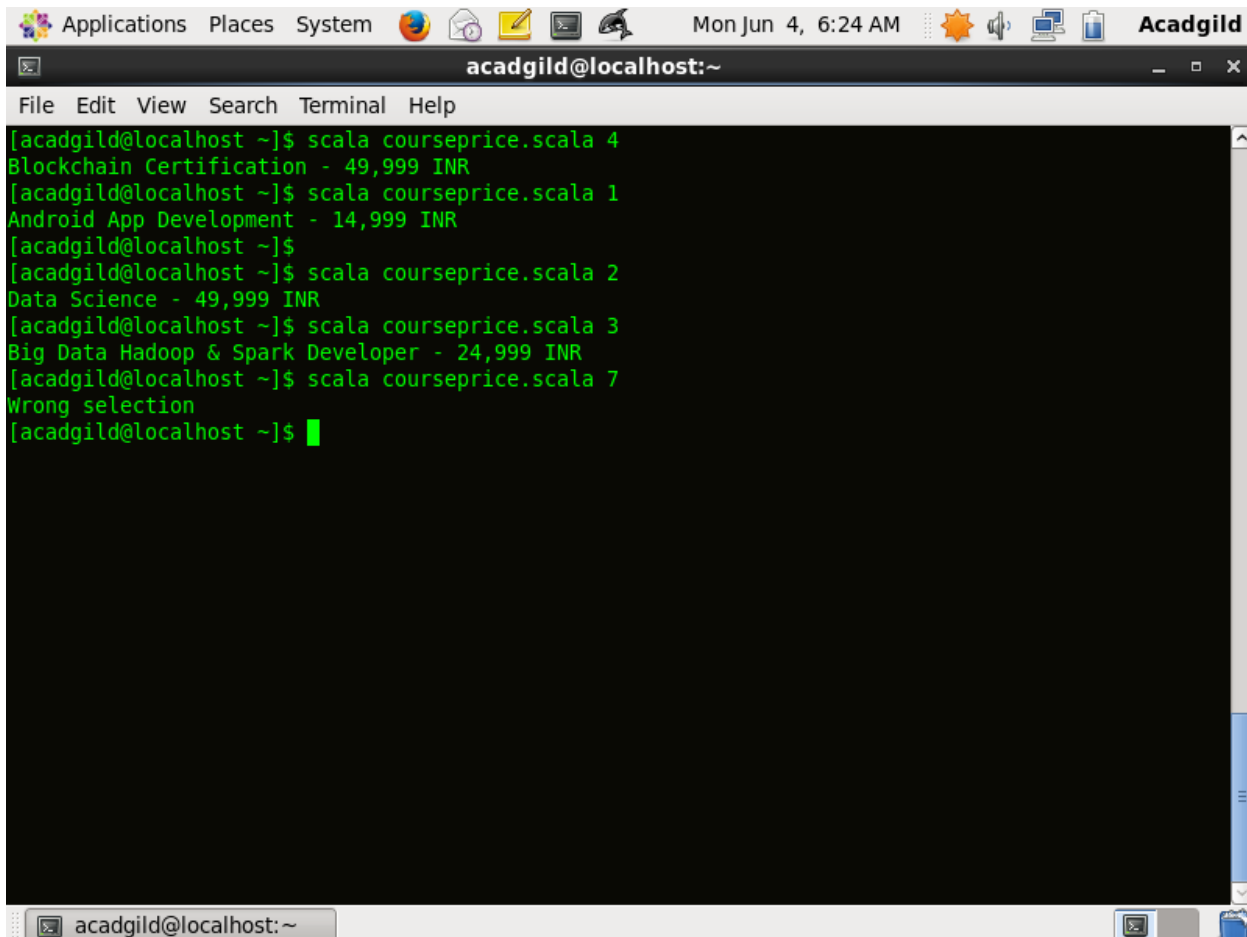
scala> getsquare(addConstantTo(2,3))
res2: Int = 6400

scala>
```

Below the terminal window, there is a gedit editor window titled '*Task_3.scala (~) - gedit'. The status bar at the bottom of the gedit window shows 'Plain Text', 'Tab Width: 8', 'Ln 8, Col 1', and 'INS'. The taskbar at the bottom of the desktop shows icons for Mozilla Firefox, the terminal window, and the gedit window.

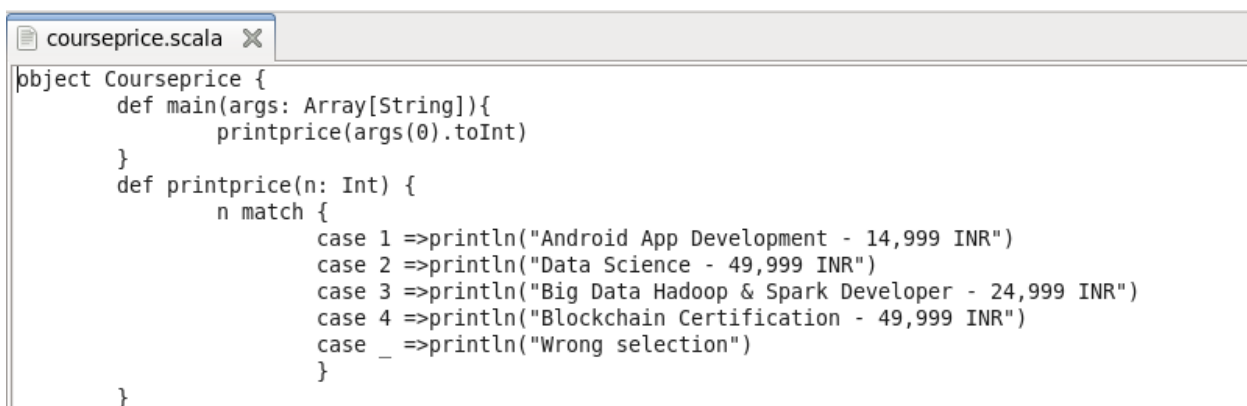
Task 4:

My script is stored in courseprice.scala file which use match to find the price of a course. The courses are numbered from 1 to 4 . The input is taken from args.



```
acadgild@localhost:~$ scala courseprice.scala 4
Blockchain Certification - 49,999 INR
[acadgild@localhost ~]$ scala courseprice.scala 1
Android App Development - 14,999 INR
[acadgild@localhost ~]$
[acadgild@localhost ~]$ scala courseprice.scala 2
Data Science - 49,999 INR
[acadgild@localhost ~]$ scala courseprice.scala 3
Big Data Hadoop & Spark Developer - 24,999 INR
[acadgild@localhost ~]$ scala courseprice.scala 7
Wrong selection
[acadgild@localhost ~]$
```

Code:



```
object Courseprice {
  def main(args: Array[String]){
    printprice(args(0).toInt)
  }
  def printprice(n: Int) {
    n match {
      case 1 =>println("Android App Development - 14,999 INR")
      case 2 =>println("Data Science - 49,999 INR")
      case 3 =>println("Big Data Hadoop & Spark Developer - 24,999 INR")
      case 4 =>println("Blockchain Certification - 49,999 INR")
      case _ =>println("Wrong selection")
    }
  }
}
```