**Assignment 15.1**

**Problem Statement**

1.Write a simple program to show inheritance in scala.

2.Write a simple program to show multiple inheritance in scala.

Solution:

1.Write a simple program to show inheritance in scala.

Inheritance in scala

Inheritance is an object oriented concept which is used to reusability of code. You can achieve inheritance by using extends keyword. To achieve inheritance a class must extend to other class. A class which is extended called super or parent class. a class which extends class is called derived or base class.

**package** com.inheritancedemo

**class** Employee{

**var** salary:Float = 10000

**def** increment = {

salary = salary + 5000

}

}

**class** Programmer **extends** Employee{

**var** bonus:Int = 5000

println("Salary = "+salary) //inherited from base class Employee

println("Bonus = "+bonus)

increment

println("New Salary = "+salary) //inherited from base class Employee

}

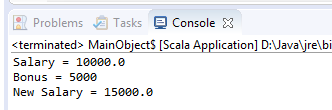
**object** MainObject{

**def** main(args:Array[*String*]){

**new** Programmer()

}

}

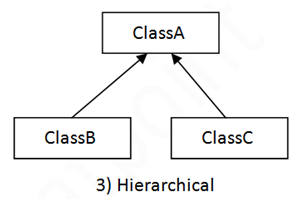
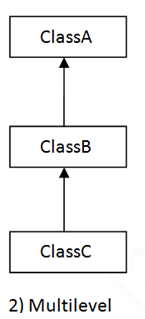
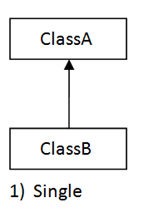


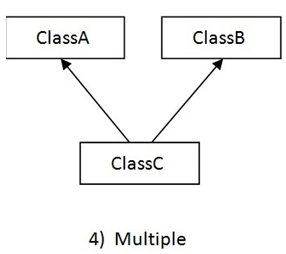
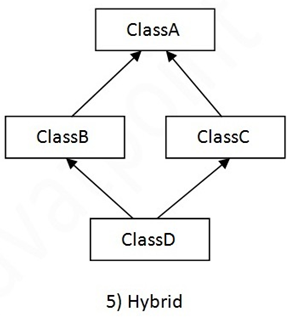
## 

2.Write a simple program to show multiple inheritance in scala.

**Types of Inheritance in Scala**

Scala supports various types of inheritance including single, multilevel, multiple, and hybrid. You can use single, multilevel and hierarchal in your class. Multiple and hybrid can only be achieved by using traits. Here, we are representing all types of inheritance by using pictorial form.



It usages scala traits concept to implement multiple inheritance. First check what is traits.

Trait a new construct defined in Scala that lies halfway between an interface and a class.

package com.inheritancedemo

**object** MultipleInheritanceExample {

**def** main(args: Array[String]): Unit = {

**trait** FirstTrait {

**var** distance: Int = \_

**def** update = {

distance = distance + 5

}

}

**trait** SecondTrait {

**var** driverVar: Int = \_

**def** update = {

driverVar = driverVar + 1

}

}

**class** FirstSecondTrait **extends** FirstTrait **with** SecondTrait {

distance = 3;

driverVar = 6;

**override** **def** update = {

**super**[FirstTrait].update

**super**[SecondTrait].update

}

}

**var** firstSceondTrait = **new** FirstSecondTrait

firstSceondTrait.update

println(firstSceondTrait.driverVar)

println(firstSceondTrait.distance)

}

}

