Name-Priyansh Salian
Class-C3
Roll No-2003148

Experiment 12: Inheritance

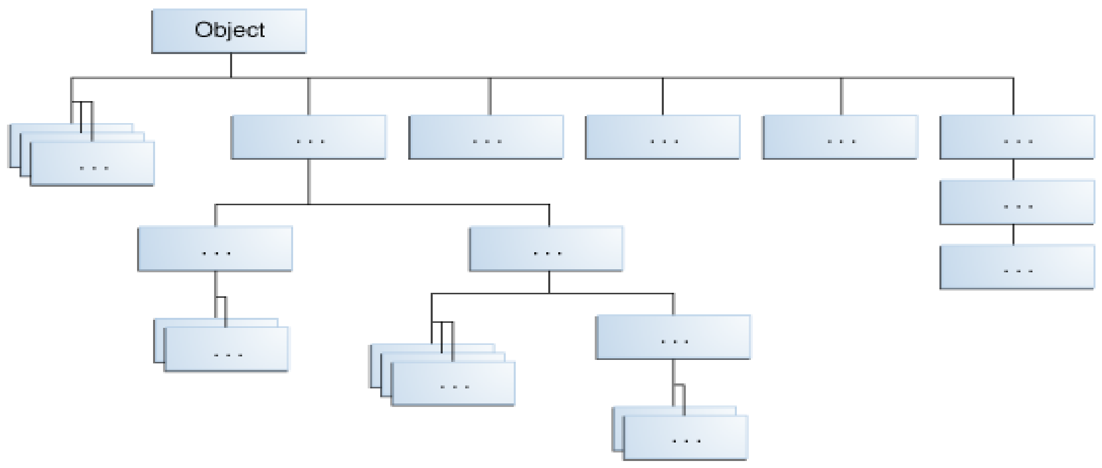
Theory :

Definitions: A class that is derived from another class is called a subclass (also a derived class, extended class, or child class). The class from which the subclass is derived is called a superclass (also a base class or a parent class).

Excepting Object, which has no superclass, every class has one and only one direct superclass (single inheritance). In the absence of any other explicit superclass, every class is implicitly a subclass of Object.

Classes can be derived from classes that are derived from classes that are derived from classes, and so on, and ultimately derived from the topmost class, Object. Such a class is said to be descended from all the classes in the inheritance chain stretching back to Object.

The idea of inheritance is simple but powerful: When you want to create a new class and there is already a class that includes some of the code that you want, you can derive your new class from the existing class. In doing this, you can reuse the fields and methods of the existing class without having to write (and debug!) them yourself.
A subclass inherits all the members (fields, methods, and nested classes) from its superclass. Constructors are not members, so they are not inherited by subclasses, but the constructor of the superclass can be invoked from the subclass.
The Java Platform Class Hierarchy
The Object class, defined in the java.lang package, defines and implements behavior common to all classes—including the ones that you write. In the Java platform, many classes derive directly from Object, other classes derive from some of those classes, and so on, forming a hierarchy of classes.

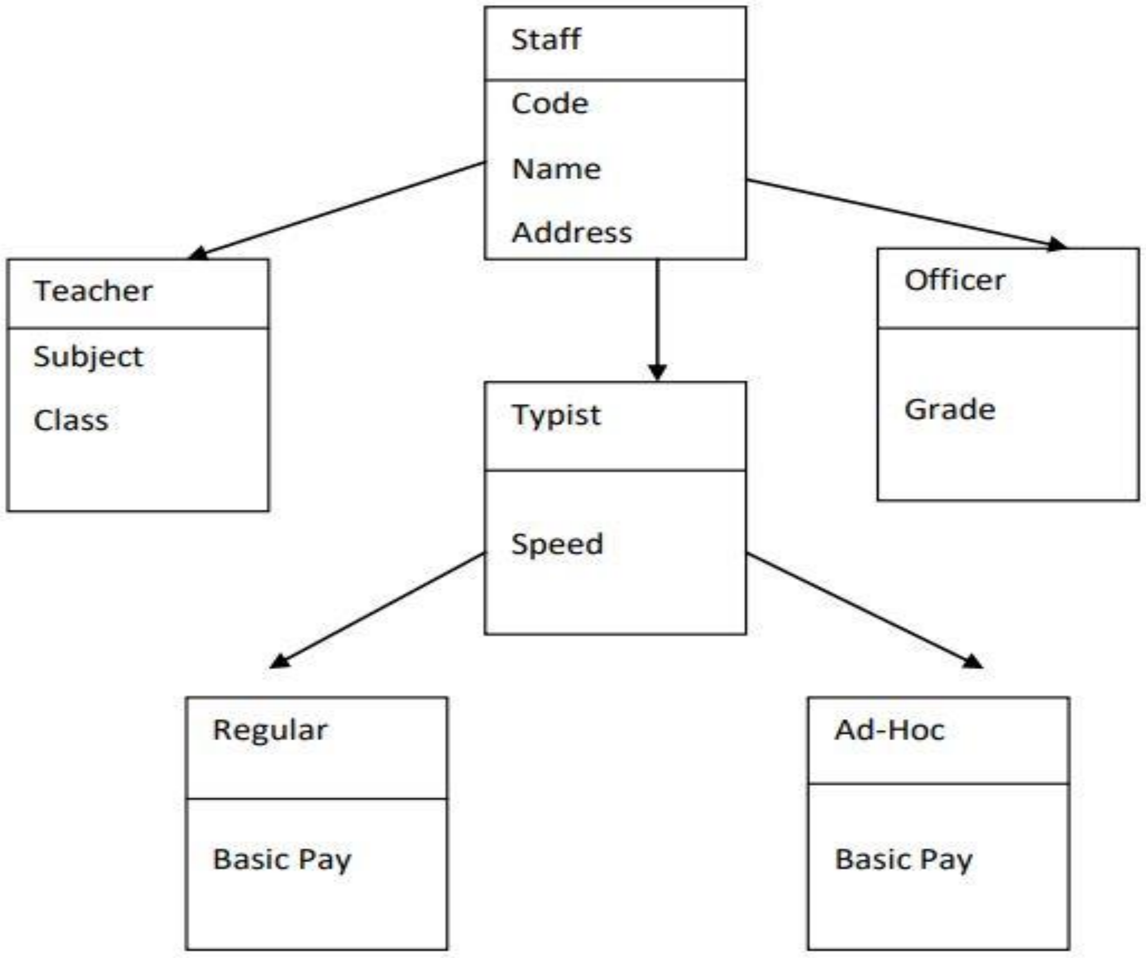


All Classes in the Java Platform are Descendants of Object

At the top of the hierarchy, Object is the most general of all classes. Classes near the bottom of the hierarchy provide more specialized behavior.

A.

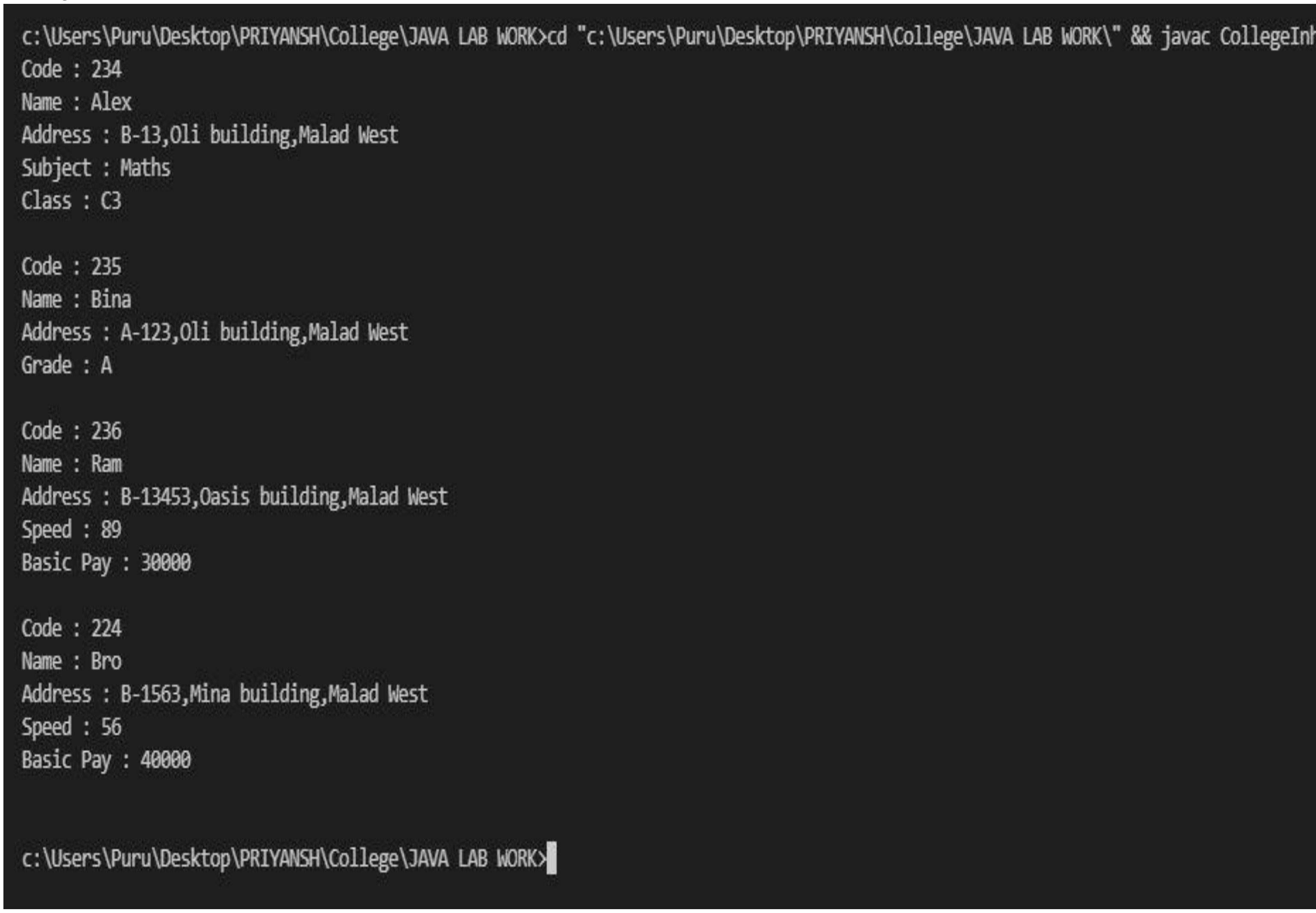
Aim : Write a Java program to implement following Inheritance



Program :

```
class Staff{
   int code;
   String name,address;

   Staff(int code,String name,String address){
      this.code=code;
      this.name=name;
      this.address=address;
   }
   void staffPrint(){
      System.out.println("Code : "+code+"\nName : "+name+"\nAddress :
"+address);
   }
}

class Teacher extends Staff{
   String subject,Class;

   Teacher(int code,String name,String address,String subject,String
Class){
      super(code, name, address);
      this.subject=subject;
      this.Class=Class;
   }
   void print(){
      staffPrint();
      System.out.println("Subject : "+subject+"\nClass : "+Class+"\n");
   }
}

class Officer extends Staff{
    String grade;

    Officer(int code,String name,String address,String grade){
        super(code, name, address);
        this.grade=grade;
    }
    void print(){
        staffPrint();
        System.out.println("Grade : "+grade+"\n");
    }
}

class Typist extends Staff{
    int speed;
    Typist(int code,String name,String address,int speed){
        super(code, name, address);
        this.speed=speed;
    }
    void typistPrint(){
        staffPrint();
        System.out.println("Speed : "+speed);
    }
}

class Regular extends Typist{
    int basic_pay;
    Regular(int code,String name,String address,int speed,int basic_pay){
        super(code, name, address, speed);
        this.basic_pay=basic_pay;
    }
    void print(){
        typistPrint();
        System.out.println("Basic Pay : "+basic_pay+"\n");
    }
}

class Ad_Hoc extends Typist{
    int basic_pay;
    Ad_Hoc(int code,String name,String address,int speed,int basic_pay){
        super(code, name, address, speed);
        this.basic_pay=basic_pay;
    }
    void print(){
        typistPrint();
        System.out.println("Basic Pay : "+basic_pay+"\n");
    }
}


public class CollegeInheritance {
    public static void main(String[] args) {
        int code,speed,basicPay;
        String name,address,subject,Class,grade;
        code=234;
        name="Alex";
        address="B-13,Oli building,Malad West";
        subject="Maths";
        Class="C3";

        Teacher t1 = new Teacher(code, name, address, subject, Class);
        code=235;
        name="Bina";
        address="A-123,Oli building,Malad West";
        subject="Maths";
        Class="C3";
        grade="A";
        Officer o1 = new Officer(code, name, address, grade);
        code=236;
        name="Ram";
        address="B-13453,Oasis building,Malad West";
        subject="Maths";
        Class="C3";
        speed=89;
        basicPay=30000;

        Regular r1 = new Regular(code, name, address, speed, basicPay);
        code=224;
        name="Bro";
        address="B-1563,Mina building,Malad West";
        subject="Maths";
        Class="C3";
        speed=56;
        basicPay=40000;

        Ad_Hoc a1 = new Ad_Hoc(code, name, address, speed, basicPay);

        t1.print();
        o1.print();
        r1.print();
        a1.print();
    }
}
```

Output :



```
c:\Users\Puru\Desktop\PRIYANSH\College\JAVA LAB WORK>cd "c:\Users\Puru\Desktop\PRIYANSH\College\JAVA LAB WORK\" && javac CollegeInh
Code : 234
Name : Alex
Address : B-13,Oli building,Malad West
Subject : Maths
Class : C3

Code : 235
Name : Bina
Address : A-123,Oli building,Malad West
Grade : A

Code : 236
Name : Ram
Address : B-13453,Oasis building,Malad West
Speed : 89
Basic Pay : 30000

Code : 224
Name : Bro
Address : B-1563,Mina building,Malad West
Speed : 56
Basic Pay : 40000


c:\Users\Puru\Desktop\PRIYANSH\College\JAVA LAB WORK>
```