Experiment Number	5
Date of Experiment	23/01/2025
Date of Submission	23/01/2025
Name of Student	Shivansh Jha
Roll Number	2330335
Section	ECSc-6

- **Title of the experiment :**

  Instructors in Java

- **Aim of The experiment :**

  To learn Java programs related to Instructors.

- **Programming Language used :**

  Java

- **Problem Statement & Solution :**

1. Modify the Java program shown in Listing 5.1 to do the following:

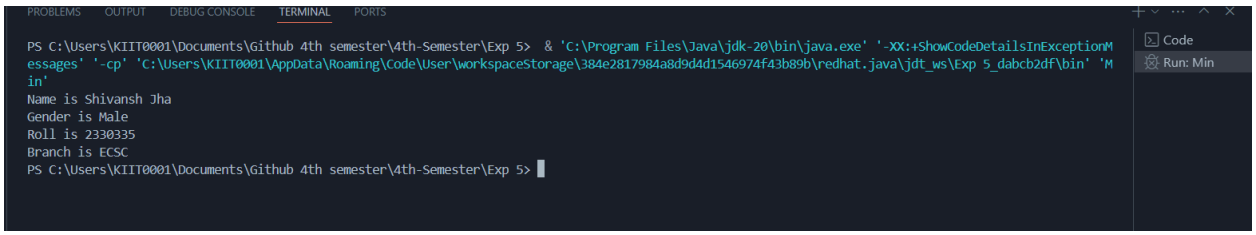
1. Add a function void gender() to class Person that prints the statement “Gender is...”

2. Add a function void branch() to class Student that prints the statement “Branch is ECSC”.

Solution :

```
class Person{
   void Gender(){
      System.out.println("Gender is Male");
   }
   void Name(){
      System.out.println("Name is Shivansh Jha");
   }
}
class Student extends Person{
   void Branch(){
      System.out.println("Branch is ECSC");
   }
   void roll(){
      System.out.println("Roll is 2330335");
   }
}
public class Min{
   public static void main(String[] args) {
      Student sc= new Student();
      sc.Name();
      sc.Gender();
      sc.roll();
      sc.Branch();
   }
}
```

Output :



2. Modify the Java program shown in Listing 5.2 to do the following:

> **1.** Add a class FirstSem that is an extension of class FirstYear. Add a function void subjects() to the class FirstSem that displays “6 theory courses and 2 lab courses”. Compile and run the program. Apply the concept of multilevel inheritance.
> **2.** Add three more classes, SecondYear, ThirdYear, FourthYear which are extensions of class Student. Add three functions, void year2(), void year3(), void year4() to three new classes SecondYear, ThirdYear, FourthYear respectively. The function void year2() displays “2nd year...”, function void year3() prints “3rd year...”, and function void year4() shows “4th year...” Apply the concept of hierarchical inheritance.

Solution :

```
class Person{
   void Name(){
      System.out.println("Name is Shivansh Jha");
   }
}
class student extends Person{
   void roll(){
      System.out.println("Roll is 2330335");
   }
}
class FirstYear extends student{
   void year(){
      System.out.println("FirstYear");
   }
}
class FirstSem extends FirstYear{
   void subjects(){
      System.out.println("6 theory courses and 2 lab courses");
   }
}
class SecondYear extends student{
   void year2(){
      System.out.println("2nd year...");
   }
}
class ThirdYear extends student{
```

```
    void year3(){
        System.out.println("3rd year...");
    }
}
class FourthYear extends student{
    void year4(){
        System.out.println("4th year...");
    }
}
public class Main {
    public static void main(String[] args) {
        FirstSem fs = new FirstSem();
        fs.Name();
        fs.roll();
        fs.year();
        fs.subjects();
        SecondYear sy = new SecondYear();
        sy.year2();
        ThirdYear ty = new ThirdYear();
        ty.year3();
        FourthYear fy = new FourthYear();
        fy.year4();
    }
}
```

Output :

```
PS C:\Users\KIIT0001\Documents\Github 4th semester\4th-Semester\Exp 5> cd "c:\Users\KIIT0001\Documents\Github 4th semester\4th-Semester\Exp 5\" ; if ($?) { javac Main.j
ava } ; if ($?) { java Main }
Name is Shivansh Jha
Roll is 2330335
FirstYear
6 theory courses and 2 lab courses
2nd year...
3rd year...
4th year...
```

3. Modify the program written for Problem 2 to do the following:

> **1.** Add variables to each of the class. Add a no-argument constructor in each of the classes to initialize the variables to default values (0 for integers and NULL/EMPTY for strings).
> **2.** Add parameterized constructors in each class to assign user inputs to the member variables.

Solution :

```
class Person1 {
   int n;
   String s;

   Person1() {
      n = 0;
      s = null;
   }

   Person1(int n, String s) {
      this.n = n;
      this.s = s;
   }
}

class student1 extends Person1 {
   int n1;
   String s1;

   student1() {
      n1 = 0;
      s1 = null;
   }

   student1(int n1, String s1) {
      this.n1 = n1;
      this.s1 = s1;
   }
```

```
}

class FirstYear1 extends student1 {
   int n1a;
   String s1a;

   FirstYear1() {
      n1a = 0;
      s1a = null;
   }

   FirstYear1(int n1a, String s1a) {
      this.n1a = n1a;
      this.s1a = s1a;
   }
}

class FirstSem1 extends FirstYear1 {
   int n1aI;
   String s1aI;
   FirstSem1() {
      n1aI = 0;
      s1aI = null;
   }
   FirstSem1(int n1aI, String s1aI){
      this.n1aI = n1aI;
      this.s1aI = s1aI;
   }
}

class SecondYear1 extends student1 {
   int n1b;
   String s1b;
   SecondYear1(){
      n1b=0;
      s1b=null;
   }
   SecondYear1(int n1b, String s1b){
      this.n1b = n1b;
      this.s1b = s1b;
```

```
    }
}

class ThirdYear1 extends student1 {
    int n2c;
    String s2c;
    ThirdYear1(){
        n2c = 0;
        s2c = null;
    }
    ThirdYear1(int n2c, String s2c){
        this.n2c=n2c;
        this.s2c=s2c;
    }
}

class FourthYear1 extends student1 {
    int n2d;
    String s2d;
    FourthYear1(){
        n2d = 0;
        s2d = null;
    }
    FourthYear1(int n2d, String s2d){
        this.n2d = n2d;
        this.s2d = s2d;
    }
}

public class Question_3 {
    public static void main(String[] args) {
        Person1 p = new Person1();
        student1 s = new student1();
        FirstYear1 fy = new FirstYear1();
        FirstSem1 fs = new FirstSem1();
        SecondYear1 sy = new SecondYear1();
        ThirdYear1 ty = new ThirdYear1();
        FourthYear1 Fy = new FourthYear1();
        System.out.println("Default value of value present in the classes");
        System.out.println("For person1 : " + p.n+" & "+p.s);
```

```
        System.out.println("For student1 : " + s.n1+" & "+s.s1);
        System.out.println("For Firstyear1 : " + fy.n1a+" & "+fy.s1a);
        System.out.println("For FirstSem1 : " + fs.n1aI+" & "+fs.s1aI);
        System.out.println("For SecondYear1 : " + sy.n1b+" & "+sy.s1b);
        System.out.println("For ThirdYear1 : " + ty.n2c+" & "+ty.s2c);
        System.out.println("For FourthYear1 : " + Fy.n2d+" & "+Fy.s2d);
        Person1 p1 = new Person1(1, "Person");
        student1 s1 = new student1(2,"Student");
        FirstYear1 fy1 = new FirstYear1(3, "FirstYear");
        FirstSem1 fs1 = new FirstSem1(4,"FirstSemester");
        SecondYear1 sy1 = new SecondYear1(5,"SecondYear");
        ThirdYear1 ty1 = new ThirdYear1(6,"ThirdYear");
        FourthYear1 Fy1 = new FourthYear1(7,"FourthYear");
        System.out.println("After initializing the value of the variable");
        System.out.println("For person1 : " + p1.n+" & "+p1.s);
        System.out.println("For student1 : " + s1.n1+" & "+s1.s1);
        System.out.println("For Firstyear1 : " + fy1.n1a+" & "+fy1.s1a);
        System.out.println("For FirstSem1 : " + fs1.n1aI+" & "+fs1.s1aI);
        System.out.println("For SecondYear1 : " + sy1.n1b+" & "+sy1.s1b);
        System.out.println("For ThirdYear1 : " + ty1.n2c+" & "+ty1.s2c);
        System.out.println("For FourthYear1 : " + Fy1.n2d+" & "+Fy1.s2d);
    }
}
```

Output :

```
PS C:\Users\KIIT0001\Documents\Github 4th semester\4th-Semester\Exp 5>  & 'C:\Program Files\Java\jdk-20\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C
:\Users\KIIT0001\AppData\Roaming\Code\User\workspaceStorage\384e2817984a8d9d4d1546974f43b89b\redhat.java\jdt_ws\Exp 5_dabcb2df\bin' 'Question_3'
Default value of value present in the classes
For person1 : 0 & null
For student1 : 0 & null
For Firstyear1 : 0 & null
For FirstSem1 : 0 & null
For SecondYear1 : 0 & null
For ThirdYear1 : 0 & null
For FourthYear1 : 0 & null
After intialzing the value of the variable
For person1 : 1 & Person
For student1 : 2 & Student
For Firstyear1 : 3 & FirstYear
For FirstSem1 : 4 & FirstSemester
For SecondYear1 : 5 & SecondYear
For ThirdYear1 : 6 & ThirdYear
For FourthYear1 : 7 & FourthYear
PS C:\Users\KIIT0001\Documents\Github 4th semester\4th-Semester\Exp 5>
```

4. Using a multi-level inheritance, write a Java program to implement the relationship shown in Figure 5.1. Also, include constructors in every class to initialize the member variables.

Solution :

```
class UniversityMember{
   String Name;char Gender;

   public UniversityMember(String Name, char Gender) {
      this.Name=Name;
      this.Gender=Gender;
   }

   public UniversityMember() {
      Name=null;
      Gender=' ';
   }

   void Display(){
      System.out.println("Name: "+Name+" Gender: "+Gender);
   }

}
class Stud extends UniversityMember{
   String course, branch;

   public Stud() {
      course=null;
      branch=null;
   }
   public Stud(String course, String branch){
      this.course=course;
      this.branch=branch;
   }
   void display(){
      System.out.println("Course: "+course+" Branch: "+branch);
   }
}
class Year extends Stud{
```

```
    int yearId, semId;
    double cgpa;
    public Year() {
        yearId=0;
        semId=0;
        cgpa=0.0;
    }
    public Year(int yearId, int semId, double cgpa){
        this.yearId=yearId;
        this.semId=semId;
        this.cgpa=cgpa;
    }
    void disp(){
        System.out.println("Year ID: "+yearId+" Semester ID: "+semId+" CGPa "+cgpa);
    }
}
public class multilevel_inheritance {
    public static void main(String[] args) {
        UniversityMember s = new UniversityMember("Shivansh Jha", 'M');
        s.Display();
        Stud st = new Stud("Btech", "CSE");
        st.display();
        Year y = new Year(1, 1, 8.0);
        y.disp();
    }
}
```

Output :

```
PS C:\Users\KIIT0001\Documents\Github 4th semester\4th-Semester\Exp 5>  c:; cd 'c:\Users\KIIT0001\Documents\Github 4th semester\4th-Semester\Exp 5'; &
'C:\Program Files\Java\jdk-20\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\KIIT0001\AppData\Roaming\Code\User\workspaceStora
ge\384e2817984a8d9d4d1546974f43b89b\redhat.java\jdt_ws\Exp 5_dabcb2df\bin' 'multilevel_inheritance'
Name: Shivansh Jha Gender: M
Course: Btech Branch: CSE
Year ID: 1 Semester ID: 1 CGPa 8.0
PS C:\Users\KIIT0001\Documents\Github 4th semester\4th-Semester\Exp 5>
```

5. Using a hierarchical inheritance, write a Java program to implement the relationship shown in Figure 5.2. Also, include constructors in every class to initialize the member variables.

Solution :

```
class UniversityMember {
    String name;
    String gender;

    UniversityMember(String name, String gender) {
        this.name = name;
        this.gender = gender;
    }

    void display() {
        System.out.println("Name: " + name);
        System.out.println("Gender: " + gender);
    }
}

class Student extends UniversityMember {
    String course;
    String branch;

    Student(String name, String gender, String course, String branch) {
        super(name, gender);
        this.course = course;
        this.branch = branch;
    }

    void displayStudentDetails() {
        display();
        System.out.println("Course: " + course);
        System.out.println("Branch: " + branch);
    }
}

class Employee extends UniversityMember {
    String employeeID;

    Employee(String name, String gender, String employeeID) {
        super(name, gender);
        this.employeeID = employeeID;
```

```
    }

    void displayEmployeeDetails() {
        display();
        System.out.println("Employee ID: " + employeeID);
    }
}

class Teaching extends Employee {
    String designation;

    Teaching(String name, String gender, String employeeID, String designation) {
        super(name, gender, employeeID);
        this.designation = designation;
    }

    void displayTeachingDetails() {
        displayEmployeeDetails();
        System.out.println("Designation: " + designation);
    }
}

class NonTeaching extends Employee {
    String designation;

    NonTeaching(String name, String gender, String employeeID, String designation) {
        super(name, gender, employeeID);
        this.designation = designation;
    }

    void displayNonTeachingDetails() {
        displayEmployeeDetails();
        System.out.println("Designation: " + designation);
    }
}

class Year extends Student {
    String yearID;
    String semID;
    String rollNumber;
```

```
    double SGPA;

    Year(String name, String gender, String course, String branch, String yearID, String semID,
String rollNumber, double SGPA) {
        super(name, gender, course, branch);
        this.yearID = yearID;
        this.semID = semID;
        this.rollNumber = rollNumber;
        this.SGPA = SGPA;
    }

    void displayYearDetails() {
        displayStudentDetails();
        System.out.println("Year ID: " + yearID);
        System.out.println("Semester ID: " + semID);
        System.out.println("Roll Number: " + rollNumber);
        System.out.println("SGPA: " + SGPA);
    }
}

public class Main {
    public static void main(String[] args) {
        Student student = new Student("John Doe", "Male", "BTech", "Computer Science");
        student.displayStudentDetails();

        Employee employee = new Employee("Jane Smith", "Female", "E12345");
        employee.displayEmployeeDetails();

        Teaching teaching = new Teaching("Dr. Alice", "Female", "T67890", "Professor");
        teaching.displayTeachingDetails();

        NonTeaching nonTeaching = new NonTeaching("Mr. Bob", "Male", "N54321", "Clerk");
        nonTeaching.displayNonTeachingDetails();

        Year year = new Year("Mike Johnson", "Male", "MTech", "Electrical Engineering", "Y2",
"S1", "R123", 9.5);
        year.displayYearDetails();
    }
}
```

Output :

- **Conclusion :**

  Learned to develop to develop Java programs using Inheritance.

Faculty Signature