St. Francis Institute of Technology, Mumbai-400 103

Department Of Information Technology

A.Y. 2020-2021

Class: SE-ITA/B, Semester: III

Subject: **Java Labs**

Experiment-3: Java Program to implement various types of inheritance.

1. **Aim:**
   i. Write a program to implement single inheritance. Declare super class 'Employee' with emp_no and emp_name.Declare subclass 'Fitness' with height and weight. Accept and display data for five employees.

   ii. Create a Teacher class and derive Professor and Associate_Professor class from Teacher class. Define appropriate constructor for all the classes. Also define a method to display information of Teacher. Make necessary assumptions as required.

2. **Prerequisite:** Knowledge of Types of Inheritances.
3. **Requirements:** Personal Computer (PC), Windows Operating System, Net beans 8.0.
4. **Pre-Experiment**
   Exercise: Theory:

   a. Inheritance:

   It allows programmers to reuse code whenever they need. Inheritance is a fundamental mechanism for building new classes from existing ones. In Java, when an "Is-A" relationship exists between two classes we use Inheritance. The parent class is termed super class and the inherited class is the sub class. The keyword "extend" is used by the sub class to inherit the features of super class. The super keyword is similar to "this" keyword. The keyword super can be used to access any data member or methods of the parent class. Super keyword can be used at variable, method and constructor level.

   b. Syntax:
   Single Inheritance:

```
class B extends A{}
```

   Multilevel Inheritance:

```
class B extends
A{} class C
extends B{}
```

Hierarchical Inheritance:

class B extends A{} class C extends A{}

5. Laboratory Exercise:

A. Procedure

i. Open Net beans for Java.
ii. Open File and Create New Java Project.
iii. Inside the Java Project rename give name to your Java Class.
iv. Click on Finish.
v. Type the Java Code in the opened class.
vi. Save the code by pressing Ctrl+S.
vii. Run the code by pressing Shift+F6.

B. Program code with comments:

Write and execute your program code to achieve the given aim and attach it **with your own comments with neat indentation**.

6. Post-Experiments Exercise

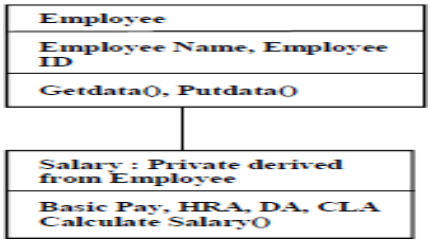
A. Extended Theory:

1. State any four points of differentiation between compile time and runtime polymorphism.
2. Explain the use of super keyword with syntax and example.

B. Results/Observations/Program output:

Present the program input/output results and comment on the same.

C. Questions/Programs:



Define classes to appropriately represent class hierarchy as shown in above figure. Use constructors for both classes and display Salary for a particular employee.

D. Conclusion:

1. Write what was performed in the experiment/program.
2. What is the significance of experiment/program?
3. Mention few applications of what was studied.

E. References

1. Balguruswamy, “Programming with java A primer”, Fifth edition, Tata McGraw Hill Publication.
2. Let Us Java-Yashwant Kanetkar.
3. Learn to Master JAVA, from Star EDU solutions , by ScriptDemics.
4. Java 8 Programming-Black Book,by-Dreamtech Publications.

Program 1:

```java
/*
Write a program to implement single inheritance. Declare super class
‘Employee’ with emp_no and emp_name.Declare subclass ‘Fitness’
with height and weight. Accept and display data for five employees.
*/

import java.util.Scanner;

class Employee{

    private int emp_no;
    private String emp_name;

    Scanner scanner = new Scanner(System.in);

    //Method to display employee name and id
    public void getData(){
        System.out.println("The name of Employee is:" + emp_name);
        System.out.println("The ID of Employee is:" + emp_no);
    }
    //Method to enter employee name and id
    public void putData(){
        System.out.println("Enter Employee name:");
        emp_name = scanner.nextLine();
        System.out.println("Enter Employee number:");
        emp_no = scanner.nextInt();
    }
}
//Class fitness extends Employee class
class Fitness extends Employee{

    private float height, weight;

    Scanner scanner = new Scanner(System.in);
    //Method to enter employee height
    public void setHeight() {
        System.out.println("Enter Employee height in centimeters:");
```

```
        height = scanner.nextFloat();
    }
    //Method to enter employee weight
    public void setWeight() {
        System.out.println("Enter Employee weight kilograms:");
        weight = scanner.nextFloat();
    }
    //Method to display employee height and weight
    public void display(){
        super.getData();
        System.out.println("The height of Employee is:" + height);
        System.out.println("The weight of Employee is:" + weight);
    }
}
public class Exp3_1 {
    public static void main(String[] args) {
        System.out.println("Enter details");
        //Instantiating object array of fitness class
        Fitness emp[] = new Fitness[5];
        //creating employee objects using constructor
        for (int i = 0; i < 5; i++) {
            emp[i] = new Fitness();
        }
        //initializing employee details
        for (int i = 0; i < 5; i++) {
            System.out.println("Employee " + (i+1));
            emp[i].putData();
            emp[i].setHeight();
            emp[i].setWeight();
            System.out.println("\n");
        }
        System.out.println("Displaying Employee Details:");
        //Displaying employee details
        for (int i = 0; i < 5; i++) {
            System.out.println("\n");
            emp[i].display();
        }
    }
}
```

Output

```
D:\College\JAVA\Experiments\Exp3>javac Exp3_1.java

D:\College\JAVA\Experiments\Exp3>java Exp3_1
Enter details
Employee 1
Enter Employee name:
John
Enter Employee number:
1
Enter Employee height in centimeters:
180
Enter Employee weight kilograms:
80


Employee 2
Enter Employee name:
Jane
Enter Employee number:
2
Enter Employee height in centimeters:
150
Enter Employee weight kilograms:
50


Employee 3
Enter Employee name:
Joseph
Enter Employee number:
3
Enter Employee height in centimeters:
160
Enter Employee weight kilograms:
60
```

```
Employee 4
Enter Employee name:
Vincent
Enter Employee number:
4
Enter Employee height in centimeters:
170
Enter Employee weight kilograms:
70


Employee 5
Enter Employee name:
Maya
Enter Employee number:
165
Enter Employee height in centimeters:
165
Enter Employee weight kilograms:
65


Displaying Employee Details:


The name of Employee is:John
The ID of Employee is:1
The height of Employee is:180.0
The weight of Employee is:80.0


The name of Employee is:Jane
The ID of Employee is:2
The height of Employee is:150.0
The weight of Employee is:50.0
```

```
The name of Employee is:Joseph
The ID of Employee is:3
The height of Employee is:160.0
The weight of Employee is:60.0


The name of Employee is:Vincent
The ID of Employee is:4
The height of Employee is:170.0
The weight of Employee is:70.0


The name of Employee is:Maya
The ID of Employee is:165
The height of Employee is:165.0
The weight of Employee is:65.0

D:\College\JAVA\Experiments\Exp3>
```

Program 2:

```java
/*
Create a Teacher class and derive Professor and Associate_Professor class
from Teacher class. Define appropriate constructor for all the classes. Also
define a method to display information of Teacher. Make necessary
assumptions as required.
*/
import java.util.Scanner;
//Defining class Teacher
class Teacher{
    private String name, dept, code;
    Scanner scanner = new Scanner(System.in);
    public void getData(){
        System.out.println("Teacher name is:" + name);
        System.out.println("Teacher department is:" + dept);
        System.out.println("Teacher subject code is:" + code);
    }
    //Constructor of class Teacher
    Teacher(){
        System.out.println("Enter teacher name :");
        name = scanner.nextLine();
        System.out.println("Enter teacher department :");
        dept = scanner.nextLine();
        System.out.println("Enter teacher subject code :");
        code = scanner.nextLine();
    }
    public String getName() {
        return name;
    }
    public String getDept() {
        return dept;
    }
    public String getCode() {
        return code;
    }
}
//Class professor inherits class Teacher
class Professor extends Teacher {
    //Constructor of class Professor
```

```
    Professor(){
        super();
        System.out.println("Professor's Information:");
        System.out.println("Name of Professor : "+getName());
        System.out.println("Professor Code : "+getCode());
        System.out.println("Department of Professor : "+getDept());
    }
}
//Class Associate_Professor inherits class Teacher
class Associate_professor extends Teacher {
    //Constructor of class associate professor
    Associate_professor(){
        super();
        System.out.println("Associate Professor's Information:");
        System.out.println("Name of Associate Professor : "+getName());
        System.out.println("Associate Professor Code : "+getCode());
        System.out.println("Department of Associate Professor : "+getDept());
    }
}
//Driver class
public class Exp3_2 {
    public static void main(String[] args) {

        //Instantiating object of class Teacher
        Teacher teacher = new Teacher();
        teacher.getData();
        new Professor();
        new Associate_professor();
    }
}
```

Output:

```
D:\College\JAVA\Experiments\Exp3>javac Exp3_2.java

D:\College\JAVA\Experiments\Exp3>java Exp3_2
Enter teacher name :
Vincent
Enter teacher department :
MECH
Enter teacher subject code :
1234
Teacher name is:Vincent
Teacher department is:MECH
Teacher subject code is:1234
Enter teacher name :
Charles
Enter teacher department :
IT
Enter teacher subject code :
5678
Professor's Information:
Name of Professor : Charles
Professor Code : 5678
Department of Professor : IT
Enter teacher name :
Josh
Enter teacher department :
CMPN
Enter teacher subject code :
7879
Associate Professor's Information:
Name of Associate Professor : Josh
Associate Professor Code : 7879
Department of Associate Professor : CMPN

D:\College\JAVA\Experiments\Exp3>
```

Questions:

Question 1:

```java
import java.util.Scanner;
//Class Emp is base class
class Emp{
    private String emp_name, emp_id;
    //Method to get employee name and id
    public void getData(){
        System.out.println("The Employee Name is: " + emp_name);
        System.out.println("The Employee ID is: " + emp_id);
    }
    //Method to set employee name and id
    public void putData(String emp_name, String emp_id){
        this.emp_name = emp_name;
        this.emp_id = emp_id;
    }
}
//Class salary inherits Emp class
class Salary extends Emp{
    private Double basic;
    //Method to get basic salary
    public Double getBasic() {
        return basic;
    }
    //Method to set basic salary
    public void setBasic(Double basic) {
        this.basic = basic;
    }
    //Method to calculate salary
    public void calculateSalary(){
        Double HRA = basic*0.3, DA = basic*0.7, CLA = 0.5*basic;
        Double salary = basic + HRA + DA + CLA;
        System.out.println("The salary of Employee is: " + salary);
    }
}
//Driver class
public class Emp_Salary{
    public static void main(String[] args) {
```

```
        Scanner scanner = new Scanner(System.in);
        String name, id;
        Double base_salary;
        System.out.println("Enter Employee name:");
        name = scanner.nextLine();
        System.out.println("Enter Employee ID:");
        id = scanner.next();
        System.out.println("Enter base Salary of Employee:");
        base_salary = scanner.nextDouble();
        //Object of employee class
        Emp emp = new Emp();
        emp.putData(name, id);
        emp.getData();
        //Object of salary class
        Salary salary = new Salary();
        salary.setBasic(base_salary);
        salary.calculateSalary();
        scanner.close();
    }
}
```

Output:

```
D:\College\JAVA\Experiments\Exp3>javac Emp_Salary.java

D:\College\JAVA\Experiments\Exp3>java Emp_Salary
Enter Employee name:
James
Enter Employee ID:
12
Enter base Salary of Employee:
250000
The Employee Name is: James
The Employee ID is: 12
The salary of Employee is: 625000.0

D:\College\JAVA\Experiments\Exp3>
```

Page No :
Date :

Josh Mahajan SE IT B 04

A] Extended Theory

1) State any four points of differentiation between compile time and runtime polymorphism.

Compile-time Polymorphism	Run-time Polymorphism
1) Compile time polymorphism means binding is occuring at compile time.	1) Runtime polymorphism means binding is occuring at runtime.
2) In compile time polymorphism call is resolved by the compiler	2) In runtime, call is not resolved by the compiler
3) It is achieved by method and operator overloading.	3) It is achieved by method overriding and virtual functions.
4) It provides fast execution because it is known early at compile time	4) It provides slow execution because it is known at run time.
5) It is less flexible	5) It is more flexible.

Exp 3

Page No :
Date :

Yash Mahajan SE IT B 04

2) Explain use of super keyword with syntax & example.

→ The superkeywords refers to superclass (parent) objects.
→ It is used to call superclass methods & to access the super class constructor.
→ The most common use of the super keyword is to eliminate the confusion between superclass & sub-class that have methods with same name.

Eg) When we have the data members of the same name in both the base & derived class, we can use the super keyword to access the member of base class, data in derived class.

```
class Base {
    int num = 30;}
class Drived extends Base {
    int num = 20;
    void callThis() {
        System.out.println("Base num: " + super.num);
        System.out.println("Drived num:" + num);
    } }
class Test {
    public static void main(String [] args) {
        Derived obj = new Drived ();
        obj.callThis ();
    } }
```

Attitude

Yash Mahajan SE IT B 04

Output :-
Base num = 30
Derived num = 20

Eg. Invoking Base class constructor.
The super keyword can also be used to invoke the parent class constructor, both parameterized & non parameterized in derived class.

```
// Base Class
class Base {
  Base (){
    System.out.println("Base class constructor");
      }
  }
// Derived Class
class ~~Derived~~ Derived extends Base {
    Derived (){
      super();
      System.out.println("Derived class constructor");
    }
  }
class Test {
  public static void main (String args[]) {
    Derived d = new Derived();
  }
}
```

Output :-
Base class constructor
Derived class constructor

Jash Mahajan SE IT B 04

0] Conclusion :-

In this Experiment we have implemented the object oriented Programming concept of Inheritance. We have written programs to study how it is achieved and what are its various types.

The aim of inheritance is to provide reusability of code so that repetition can be avoided.

Inheritance mainly has two important uses, first is to reuse code to avoid repetition of code and overriding which allows a sub-class to provide specific implementation of code that already exists.