

**(Established under the Presidency University Act, 2013 of the Karnataka Act 41 of 2013)**

**Course Code : CSE1006 Course Title : Problem solving using java**

‘super’ keyword in java:

1. **Access Parent Class Variables:** If a variable in the subclass **hides** a variable in the parent class, use **super.variableName** to access the parent’s version.
2. **Call Parent Class Methods:** If the child class **overrides** a method from the parent class, you can still call the parent’s method using **super.methodName().**
3. **Call Parent Class Constructor:** Use super(values) to call the **parent class constructor** from the child class constructor. This must be the **first statement** in the child class constructor.

1.. You are building a school application. You have a Student class that stores name and roll number. Create a derived class Marks that adds subject marks. Write code to input and display student info and marks using single level inheritance.

import java.util.Scanner;

// Base class

class Student {

String name;

int rollNo;

Student(String n, int r)// base or parent or super class constructor

{

name = n;

rollNo = r;

}

void displayStudent() {

System.out.println("Name: " + name);

System.out.println("Roll No: " + rollNo);

}

}

// Derived class

class Marks extends Student {

int javaMarks, chemMarks;

Marks(String n, int r, int j, int c)// child or sub or derived class

// constructor

{

super(n,r); // calling parent class constructor

javaMarks = j;

chemMarks = c;

}

void displayMarks() {

displayStudent(); // calling student class method

System.out.println("java marks : " + javaMarks);

System.out.println("chemistry marks : " + chemMarks);

int total=javaMarks+chemMarks;

System.out.println("Total : " + total);

}

}

public class SingleLevel {

public static void main(String[] args) {

Marks m = new Marks("suhas",111,70,65);

System.out.println("Student Information ");

m.displayMarks();

}

}

===============

2.Rewrite the above program using method overriding

import java.util.Scanner;

// Base class

class Student {

String name;

int rollNo;

Student(String n, int r) //base or parent or super class constructor

{

name = n;

rollNo = r;

}

void display() {

System.out.println("Name: " + name);

System.out.println("Roll No: " + rollNo);

}

}

// Derived class

class Marks extends Student {

int javaMarks, chemMarks;

Marks(String n, int r, int j, int c)// child or sub or derived class

//constructor

super(n,r); // call to parent class constructor

javaMarks = j;

chemMarks = c;

}

void display() {

super.display(); // calling parent class display method

System.out.println("java marks : " + javaMarks);

System.out.println("chemistry marks : " + chemMarks);

int total=javaMarks+chemMarks;

System.out.println("Total : " + total);

}

}

public class SingleLevel3 {

public static void main(String[] args) {

Marks m = new Marks("suhas",111,70,65);

System.out.println("Student Information ");

m.display();

}

}

================

3. You are building a school application. You have a Student class that stores name and roll number. Create a derived class Marks that adds subject marks. Create a derived class Practicals that adds both theory and Lab marks. Write code to input and display student info and marks using multi level inheritance.

import java.util.Scanner;

// Base class

class Student {

protected String name;

protected int rollNo;

Student(String n, int r)// base or parent or super class constructor

{

name = n;

rollNo = r;

}

public void display() {

System.out.println("Name: " + name);

System.out.println("Roll No: " + rollNo);

}

}

// Derived class

class Marks extends Student {

protected int javaMarks, chemMarks;

Marks(String n, int r, int j, int c) { // child class

//constructor

super(n,r); // call to parent class constructor

javaMarks = j;

chemMarks = c;

}

public void display() {

super.display(); // calling parent class display method

System.out.println("java marks : " + javaMarks);

System.out.println("chemistry marks : " + chemMarks);

int total=javaMarks+chemMarks;

System.out.println("Total : " + total);

}

}

class Practicals extends Marks{

int javaLab, chemLab;

Practicals(String n, int r, int j, int c, int javaLab, int chemLab)

{

super(n,r,j,c); // calling Marks class constructor

this.javaLab=javaLab;

this.chemLab=chemLab;

}

public void display() {

super.display(); // calling Marks class display

System.out.println("java Lab marks : " + javaLab);

System.out.println("chemistry Lab marks : " + chemLab);

int grandTotal=javaMarks+chemMarks+javaLab+chemLab;

System.out.println("Grand Total : " + grandTotal);

}

}

public class MultiLevel1{

public static void main(String[] args) {

Practicals p = new Practicals("suhas",111,50,60,70,80);

System.out.println("Student Information ");

p.display(); // calling Practical class display

}

}

================

4. demo of super keyword to access parent class member

// base class vehicle

class Vehicle

{

int maxSpeed = 140;

}

// sub class Car extending vehicle

class Car extends Vehicle

{

int maxSpeed = 180;

void display()

{

System.out.println("Maximum Speed of the vehicle" + super.maxSpeed);

System.out.println("Maximum Speed of the car " + maxSpeed);

}

}

class Main

{

public static void main(String[] args)

{

Car c = new Car();

c.display();

}

}

===============

**5. Java program** that demonstrates **hierarchical inheritance** using Person as the **superclass**, and Employee and Student as **subclasses**. This structure shows how different classes can inherit common properties from a single parent class.

// Superclass

class Person {

String name;

int age;

void setPersonDetails(String name, int age) {

this.name = name;

this.age = age;

}

void getPersonDetails() {

System.out.println("Name: " + name);

System.out.println("Age: " + age);

}

}

// Subclass 1

class Employee extends Person {

String company;

double salary;

void setEmployeeDetails(String company, double salary) {

this.company = company;

this.salary = salary;

}

void getEmployeeDetails() {

getPersonDetails();

System.out.println("Company: " + company);

System.out.println("Salary: " + salary);

}

}

// Subclass 2

class Student extends Person {

String school;

double cgpa;

void setStudentDetails(String school, double cgpa) {

this.school = school;

this.cgpa = cgpa;

}

void getStudentDetails() {

getPersonDetails();

System.out.println("School: " + school);

System.out.println("cgpa : " + cgpa);

}

}

// Main class

public class Main {

public static void main(String[] args) {

// Employee object

Employee emp = new Employee();

emp.setPersonDetails("Darshan", 30);

emp.setEmployeeDetails("Accenture", 95000);

System.out.println("Employee Info :");

emp.getEmployeeDetails();

// Student object

Student stu = new Student();

stu.setPersonDetails("Rithika", 19);

stu.setStudentDetails("Presidency", 8.5);

System.out.println("Student Info:");

stu.getStudentDetails();

}

}

================

Activities for the students:

1. Rewrite program 5, using constructors (instead of set methods)
2. Rewrite program 2, without using constructors (that is write set or getdata() methods)

================