1.Single Inheritance

class Teacher {

private String name;

private String subject;

public Teacher(String name, String subject) {

this.name = name;

this.subject = subject;

}

public void displayInfo() {

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

System.out.println("Subject: " + subject);

}

}

class MathTeacher extends Teacher {

private int numberOfClasses;

public MathTeacher(String name, String subject, int numberOfClasses) {

super(name, subject);

this.numberOfClasses = numberOfClasses;

}

public void displayMathTeacherInfo() {

displayInfo();

System.out.println("Number of Classes: " + numberOfClasses);

}

}

public class Main {

public static void main(String[] args) {

MathTeacher mt = new MathTeacher("Mr. Sharma", "Mathematics", 5);

mt.displayMathTeacherInfo();

}

}

2.Multilevel Inheritance

class Employee {

protected String name;

protected double salary;

public Employee(String name, double salary) {

this.name = name;

this.salary = salary;

}

public void displayEmployee() {

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

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

}

}

class Manager extends Employee {

protected String department;

public Manager(String name, double salary, String department) {

super(name, salary);

this.department = department;

}

public void displayManager() {

displayEmployee();

System.out.println("Department: " + department);

}

}

class SeniorManager extends Manager {

private int teamSize;

public SeniorManager(String name, double salary, String department, int teamSize) {

super(name, salary, department);

this.teamSize = teamSize;

}

public void displaySeniorManager() {

displayManager();

System.out.println("Team Size: " + teamSize);

}

}

public class Main {

public static void main(String[] args) {

SeniorManager sm = new SeniorManager("Alice", 95000, "IT", 12);

sm.displaySeniorManager();

}

}

3.Hierarchical Inheritance

class Animal {

protected String name;

protected int age;

public Animal(String name, int age) {

this.name = name;

this.age = age;

}

public void displayInfo() {

System.out.println("Animal Name: " + name + ", Age: " + age);

}

}

class Lion extends Animal {

public Lion(String name, int age) {

super(name, age);

}

public void roar() {

System.out.println(name + " roars loudly! 🦁");

}

}

class Elephant extends Animal {

public Elephant(String name, int age) {

super(name, age);

}

public void trumpet() {

System.out.println(name + " trumpets with its trunk! 🐘");

}

}

class Monkey extends Animal {

public Monkey(String name, int age) {

super(name, age);

}

public void climb() {

System.out.println(name + " climbs trees swiftly! 🐒");

}

}

public class Main {

public static void main(String[] args) {

Lion lion = new Lion("Simba", 5);

Elephant elephant = new Elephant("Dumbo", 10);

Monkey monkey = new Monkey("George", 3);

lion.displayInfo();

lion.roar();

elephant.displayInfo();

elephant.trumpet();

monkey.displayInfo();

monkey.climb();

}

}

4.Hybrid Inheritance

class User {

protected String name;

protected String email;

public User(String name, String email) {

this.name = name;

this.email = email;

}

public void displayUser() {

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

}

}

class Customer extends User {

private int customerId;

public Customer(String name, String email, int customerId) {

super(name, email);

this.customerId = customerId;

}

public void displayCustomer() {

displayUser();

System.out.println("Customer ID: " + customerId);

}

}

class Seller extends User {

private String shopName;

public Seller(String name, String email, String shopName) {

super(name, email);

this.shopName = shopName;

}

public void displaySeller() {

displayUser();

System.out.println("Shop Name: " + shopName);

}

}

interface PrimeFeatures {

void freeDelivery();

void exclusiveDeals();

}

class PrimeCustomer extends Customer implements PrimeFeatures {

private String primeId;

public PrimeCustomer(String name, String email, int customerId, String primeId) {

super(name, email, customerId);

this.primeId = primeId;

}

@Override

public void freeDelivery() {

System.out.println("PrimeCustomer " + name + " gets FREE delivery!");

}

@Override

public void exclusiveDeals() {

System.out.println("PrimeCustomer " + name + " gets access to EXCLUSIVE deals!");

}

public void displayPrimeCustomer() {

displayCustomer();

System.out.println("Prime ID: " + primeId);

}

}

public class Main {

public static void main(String[] args) {

Seller seller = new Seller("Rahul", "rahul@sellers.com", "Rahul Electronics");

System.out.println("\n--- Seller Details ---");

seller.displaySeller();

PrimeCustomer primeCust = new PrimeCustomer("Aisha", "aisha@prime.com", 101, "PR123");

System.out.println("\n--- Prime Customer Details ---");

primeCust.displayPrimeCustomer();

primeCust.freeDelivery();

primeCust.exclusiveDeals();

}

}

5. Super keyword

class Bank {

protected double interestRate = 5.0;

public void displayRate() {

System.out.println("Bank Interest Rate: " + interestRate + "%");

}

}

class SBI extends Bank {

protected double interestRate = 6.5;

@Override

public void displayRate() {

System.out.println("SBI Interest Rate: " + interestRate + "%");

System.out.println("Parent Bank Interest Rate using super: " + super.interestRate + "%");

super.displayRate();

}

}

public class Main {

public static void main(String[] args) {

SBI sbi = new SBI();

sbi.displayRate();

}

}