**Inheritance**

1. Create a class named 'Member' having the following members:

Data members  
1 - Name  
2 - Age  
3 - Phone number  
4 - Address  
5 - Salary  
It also has a method named 'printSalary' which prints the salary of the members.  
Two classes 'Employee' and 'Manager' inherits the 'Member' class. The 'Employee' and 'Manager' classes have data members 'specialization' and 'department' respectively. Now, assign name, age, phone number, address and salary to an employee and a manager by making an object of both of these classes and print the same.

**CODE:**

import java.util.Scanner;

class Member {

String name;

int age;

String phoneNumber;

String address;

double salary;

public Member(String name, int age, String phoneNumber, String address, double salary) {

this.name = name;

this.age = age;

this.phoneNumber = phoneNumber;

this.address = address;

this.salary = salary;

}

public void printSalary() {

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

}

}

class Employee extends Member {

String specialization;

public Employee(String name, int age, String phoneNumber, String address, double salary, String specialization) {

super(name, age, phoneNumber, address, salary);

this.specialization = specialization;

}

}

class Manager extends Member {

String department;

public Manager(String name, int age, String phoneNumber, String address, double salary, String department) {

super(name, age, phoneNumber, address, salary);

this.department = department;

}

}

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter Employee details:");

System.out.print("Name: ");

String employeeName = scanner.nextLine();

System.out.print("Age: ");

int employeeAge = scanner.nextInt();

scanner.nextLine();

System.out.print("Phone Number: ");

String employeePhoneNumber = scanner.nextLine();

System.out.print("Address: ");

String employeeAddress = scanner.nextLine();

System.out.print("Salary: ");

double employeeSalary = scanner.nextDouble();

scanner.nextLine();

System.out.print("Specialization: ");

String employeeSpecialization = scanner.nextLine();

Employee employee = new Employee(employeeName, employeeAge, employeePhoneNumber, employeeAddress, employeeSalary, employeeSpecialization);

System.out.println("\nEnter Manager details:");

System.out.print("Name: ");

String managerName = scanner.nextLine();

System.out.print("Age: ");

int managerAge = scanner.nextInt();

scanner.nextLine();

System.out.print("Phone Number: ");

String managerPhoneNumber = scanner.nextLine();

System.out.print("Address: ");

String managerAddress = scanner.nextLine();

System.out.print("Salary: ");

double managerSalary = scanner.nextDouble();

scanner.nextLine();

System.out.print("Department: ");

String managerDepartment = scanner.nextLine();

Manager manager = new Manager(managerName, managerAge, managerPhoneNumber, managerAddress, managerSalary, managerDepartment);

scanner.close();

System.out.println("\nEmployee Details:");

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

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

System.out.println("Phone Number: " + employee.phoneNumber);

System.out.println("Address: " + employee.address);

employee.printSalary();

System.out.println("Specialization: " + employee.specialization);

System.out.println();

System.out.println("Manager Details:");

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

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

System.out.println("Phone Number: " + manager.phoneNumber);

System.out.println("Address: " + manager.address);

manager.printSalary();

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

}

}

2. You are developing a banking application in Java. Design a class hierarchy that represents different account types such as SavingsAccount, CheckingAccount, and LoanAccount.

Each account should have basic functionality like deposit, withdraw, and check balance.

Ensure that your design follows appropriate use of interfaces and inheritance.

**CODE:**

import java.util.Scanner;

interface Account {

void deposit(double amount);

void withdraw(double amount);

double checkBalance();

}

interface InterestBearing {

void calculateInterest();

}

class BaseAccount implements Account {

protected double balance;

public void deposit(double amount) {

if (amount > 0) {

balance += amount;

System.out.println("Deposit of $" + amount + " successful.");

} else {

System.out.println("Invalid deposit amount.");

}

}

public void withdraw(double amount) {

if (amount > 0 && balance >= amount) {

balance -= amount;

System.out.println("Withdrawal of $" + amount + " successful.");

} else {

System.out.println("Invalid withdrawal amount or insufficient funds.");

}

}

public double checkBalance() {

return balance;

}

}

class SavingsAccount extends BaseAccount implements InterestBearing {

private double interestRate;

public SavingsAccount(double initialBalance, double interestRate) {

this.balance = initialBalance;

this.interestRate = interestRate;

}

public void calculateInterest() {

double interest = balance \* interestRate / 100;

balance += interest;

System.out.println("Interest calculated and added to the account.");

}

}

class CheckingAccount extends BaseAccount {

public CheckingAccount(double initialBalance) {

this.balance = initialBalance;

}

}

class LoanAccount extends BaseAccount {

private double interestRate;

public LoanAccount(double initialBalance, double interestRate) {

this.balance = initialBalance;

this.interestRate = interestRate;

}

public void withdraw(double amount) {

balance -= amount;

System.out.println("Withdrawal from loan account of $" + amount + " successful.");

}

public void calculateInterest() {

double interest = balance \* interestRate / 100;

balance += interest;

System.out.println("Interest calculated and added to the loan account.");

}

}

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter initial balance for Savings Account:");

double savingsInitialBalance = scanner.nextDouble();

System.out.println("Enter interest rate for Savings Account:");

double savingsInterestRate = scanner.nextDouble();

SavingsAccount savingsAccount = new SavingsAccount(savingsInitialBalance, savingsInterestRate);

System.out.println("Enter initial balance for Checking Account:");

double checkingInitialBalance = scanner.nextDouble();

CheckingAccount checkingAccount = new CheckingAccount(checkingInitialBalance);

System.out.println("Enter initial balance for Loan Account:");

double loanInitialBalance = scanner.nextDouble();

System.out.println("Enter interest rate for Loan Account:");

double loanInterestRate = scanner.nextDouble();

LoanAccount loanAccount = new LoanAccount(loanInitialBalance, loanInterestRate);

System.out.println("\nOperations on Savings Account:");

System.out.print("Enter deposit amount: $");

double savingsDepositAmount = scanner.nextDouble();

savingsAccount.deposit(savingsDepositAmount);

System.out.print("Enter withdrawal amount: $");

double savingsWithdrawAmount = scanner.nextDouble();

savingsAccount.withdraw(savingsWithdrawAmount);

System.out.println("Savings Account Balance: $" + savingsAccount.checkBalance());

System.out.println("\nOperations on Checking Account:");

System.out.print("Enter deposit amount: $");

double checkingDepositAmount = scanner.nextDouble();

checkingAccount.deposit(checkingDepositAmount);

System.out.print("Enter withdrawal amount: $");

double checkingWithdrawAmount = scanner.nextDouble();

checkingAccount.withdraw(checkingWithdrawAmount);

System.out.println("Checking Account Balance: $" + checkingAccount.checkBalance());

System.out.println("\nOperations on Loan Account:");

System.out.print("Enter withdrawal amount: $");

double loanWithdrawAmount = scanner.nextDouble();

loanAccount.withdraw(loanWithdrawAmount);

loanAccount.calculateInterest();

System.out.println("Loan Account Balance: $" + loanAccount.checkBalance());

scanner.close();

}

}

3. You are tasked with designing a university enrollment system in Java. Implement a class hierarchy that includes a base class **Person**and two subclasses, **Student**and **Professor**and a**Course**class. Each class should have the necessary attributes. Each course should have a list of prerequisites and enrolled students.

Your tasks are as follows:

i) Students should only be enrolled if they have completed all the required prerequisites. In the course class, include logic for enrolling students.

ii) Display enrolled students in a particular with relevant information.

**CODE:**

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

class Person {

protected String name;

public Person(String name) {

this.name = name;

}

}

class Student extends Person {

public Student(String name) {

super(name);

}

}

class Professor extends Person {

public Professor(String name) {

super(name);

}

}

class Course {

private String courseName;

private List<String> prerequisites;

private List<Student> enrolledStudents;

public Course(String courseName, List<String> prerequisites) {

this.courseName = courseName;

this.prerequisites = prerequisites;

this.enrolledStudents = new ArrayList<>();

}

public void enrollStudent(Scanner scanner) {

System.out.println("Enter student name:");

String studentName = scanner.nextLine();

Student student = new Student(studentName);

if (hasCompletedPrerequisites(scanner, student)) {

enrolledStudents.add(student);

System.out.println("Enrollment successful for " + student.name + " in course " + courseName);

} else {

System.out.println("Enrollment failed for " + student.name + " in course " + courseName +

". Prerequisites not met.");

}

}

private boolean hasCompletedPrerequisites(Scanner scanner, Student student) {

for (String prerequisite : prerequisites) {

System.out.println(student.name + ", have you completed prerequisite: " + prerequisite + "? (yes/no)");

String response = scanner.nextLine().toLowerCase();

if (!response.equals("yes")) {

return false;

}

}

return true;

}

public void displayEnrolledStudents() {

System.out.println("Enrolled students in course " + courseName + ":");

for (Student student : enrolledStudents) {

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

}

}

}

public class Main {

public static void main(String[] args) {

List<String> prerequisites = new ArrayList<>();

prerequisites.add("c");

prerequisites.add("Python");

Course course = new Course("java", prerequisites);

try (Scanner scanner = new Scanner(System.in)) {

course.enrollStudent(scanner);

course.enrollStudent(scanner);

}

course.displayEnrolledStudents();

}

}