

Lab 06 Tasks

Task 01

A company wants to manage its employees and identify managers separately. Each employee has a name and salary, while managers receive an additional bonus. The HR department needs a program to store this information and display employee details.

Classes & Members

1. Base Class: Employee

- **Data Members:** name (string), salary (float)
- **Member Functions:**
 - Constructor to initialize name and salary
 - displayDetails() to show employee details

2. Derived Class: Manager (inherits from Employee)

- **Data Member:** bonus (float)
- **Member Functions:**
 - Constructor to initialize all attributes (name, salary, bonus)
 - displayDetails() (overrides to include bonus)

Program Flow

- HR inputs employee details for a manager.
- The program displays name, salary, and bonus.

Task 02

An automobile company wants a system to store details of their vehicles. They manufacture different types of cars, including electric cars. Each vehicle has a brand and speed, a car has seating capacity, and an electric car has battery life.

Classes & Members

1. Base Class: Vehicle

- **Data Members:** brand (string), speed (int)
- **Member Functions:** displayDetails()

2. Derived Class: Car

- **Data Member:** seats (int)
- **Member Functions:** displayDetails()

3. Derived Class: ElectricCar

- **Data Member:** batteryLife (int)
- **Member Functions:** displayDetails()

Program Flow

- A car showroom enters details of an electric car.
- The program displays brand, speed, seats, and batteryLife.

Task 03

A university wants a system to manage its professors, who are both teachers and researchers. Every professor has a name, age, subject expertise, research area, and publications.

Classes & Members

1. **Base Class: Person**
 - **Data Members:** name (string), age (int)
 - **Member Functions:** displayDetails()
2. **Derived Class: Teacher**
 - **Data Member:** subject (string)
 - **Member Functions:** displayDetails()
3. **Derived Class: Researcher**
 - **Data Member:** researchArea (string)
 - **Member Functions:** displayDetails()
4. **Derived Class: Professor**
 - **Data Member:** publications (int)
 - **Member Functions:** displayDetails()

Program Flow

- The university enters details of a professor.
- The program displays name, age, subject, researchArea, and publications.

Task 04

A bank wants a system to manage different types of customer accounts. They offer savings and checking accounts. A savings account earns interest, while a checking account has an overdraft limit.

Classes & Members

1. **Base Class: Account**
 - **Data Members:** accountNumber (int), balance (float)

- **Member Functions:** displayDetails()
- 2. **Derived Class: SavingsAccount**
 - **Data Member:** interestRate (float)
 - **Member Functions:** displayDetails()
- 3. **Derived Class: CheckingAccount**
 - **Data Member:** overdraftLimit (float)
 - **Member Functions:** displayDetails()

Program Flow

- A bank employee enters details for both types of accounts.
- The program displays accountNumber, balance, interestRate, or overdraftLimit.

Task 05

A tech company is developing a new **smart wearable device** that combines features of a **smartphone** and a **smartwatch**. This device has a **screen size**, a **heart rate monitor**, and a **step counter**.

Classes & Members

1. **Base Class: Device**
 - **Data Members:** deviceID (int), status (bool)
 - **Member Functions:** displayDetails()
2. **Derived Class: SmartPhone**
 - **Data Member:** screenSize (float)
 - **Member Functions:** displayDetails()
3. **Derived Class: SmartWatch**
 - **Data Member:** heartRateMonitor (bool)
 - **Member Functions:** displayDetails()
4. **Derived Class: SmartWearable**
 - **Data Member:** stepCounter (int)
 - **Member Functions:** displayDetails()

Program Flow

- A developer enters the specifications of the smart wearable.
- The program displays all attributes: deviceID, status, screenSize, heartRateMonitor, and stepCounter.