🚖

CPP-A19: Abstraction

📝 Problem 1: Smart Home Device Controller

📌 Problem Statement:

Design a **Smart Home System** where different devices (Light, Fan, AC) can be turned **ON/OFF**. The system should provide a way to:

- **Turn ON/OFF any device** using a standard function.
- Each device should have its own behavior when turned ON or OFF.

🔷 **Hint:** Use **Abstraction** through an **Abstract Class** `Device`.

🔻 **Example Input & Output:**

```
Input:
Device* d1 = new Light();
Device* d2 = new Fan();
Device* d3 = new AC();
d1→turnOn();
d2→turnOn();
d3→turnOn();

Output:
Light is now ON.
Fan is now rotating.
AC is now cooling.
```

📝 Problem 2: Secure Payment System

📌 Problem Statement:

Create a **payment processing system** where multiple payment methods are supported:

1. **Credit Card**
2. **PayPal**
3. **UPI**

Each payment method should implement a `processPayment(double amount)` function.

🔷 **Hint:** Use an **Abstract Class** `PaymentMethod`.

🔻 **Example Input & Output:**

```
Input:
PaymentMethod* p1 = new CreditCard();
p1->processPayment(500);

Output:
Processing Credit Card payment of $500.
```

📝 Problem 3: Employee Payroll System

📌 **Problem Statement:**

Implement an **Employee Payroll System** where different types of employees have different salary structures:

- `FullTimeEmployee` (Fixed Salary)
- `PartTimeEmployee` (Hourly Wage)
- `Freelancer` (Project-Based Payment)

🔷 **Hint:** Use an **Abstract Class** `Employee` with a `calculateSalary()` function.

🔻 **Example Input & Output:**

```
Input:
Employee* e1 = new FullTimeEmployee(50000);
Employee* e2 = new PartTimeEmployee(40, 20);
Employee* e3 = new Freelancer(3, 2000);
e1->calculateSalary();
e2->calculateSalary();
e3->calculateSalary();

Output:
```

```
Full-Time Employee Salary: $50000
Part-Time Employee Salary: $800
Freelancer Payment: $6000
```

📝 Problem 4: Online Learning Platform

📌 Problem Statement:

Design a **Learning Management System** where different types of courses exist:

1. **VideoCourse** (Video Duration-Based)
2. **LiveClass** (Scheduled Timings)
3. **QuizCourse** (Interactive Quiz-Based)

Each course should have a `showCourseDetails()` function.

🔷 **Hint:** Use an **Abstract Class** `Course`.

🔻 Example Input & Output:

```
Input:
Course* c1 = new VideoCourse("C++ Basics", 3);
Course* c2 = new LiveClass("DSA", "10 AM - 12 PM");
c1->showCourseDetails();
c2->showCourseDetails();

Output:
Video Course: C++ Basics, Duration: 3 hours
Live Class: DSA, Timing: 10 AM - 12 PM
```

📝 Problem 5: Vehicle Rental System

📌 Problem Statement:

Create a **Vehicle Rental System** where different types of vehicles can be rented:

- `Car`
- `Bike`

- `Truck`

Each vehicle should have a `rentVehicle(int hours)` function to calculate rental cost based on per-hour charges.

🔷 **Hint:** Use an **Abstract Class** `Vehicle`.

🔻 **Example Input & Output:**

```
Input:
Vehicle* v1 = new Car();
Vehicle* v2 = new Bike();
v1->rentVehicle(5);
v2->rentVehicle(3);

Output:
Car rented for 5 hours. Total cost: $250
Bike rented for 3 hours. Total cost: $90
```

📝 Problem 6: Airline Reservation System

📌 Problem Statement:

Design an **Airline Reservation System** where different types of seats are available:

1. **Economy Class**
2. **Business Class**
3. **First Class**

Each seat type should have its own method to **calculate the ticket price** based on base fare and additional charges.

🔷 **Hint:** Use an **Abstract Class** `Seat` with a `calculateFare(double baseFare)` function.

🔻 **Example Input & Output:**

```
Input:
Seat* s1 = new EconomyClass();
Seat* s2 = new BusinessClass();
s1->calculateFare(500);
s2->calculateFare(500);
```

```
Output:
Economy Class Fare: $500
Business Class Fare: $750
```

📝 Problem 7: Smart Parking System

📌 Problem Statement:

Create a **Smart Parking System** that can handle different vehicle types:

1. **Car**
2. **Bike**
3. **Truck**

Each vehicle type has a different parking charge per hour. Implement a function to calculate the **total parking fee** based on the number of hours parked.

🔷 **Hint:** Use an **Abstract Class** `ParkingSpot`.

🔻 Example Input & Output:

```
Input:
ParkingSpot* p1 = new CarParking();
p1→calculateFee(3);

Output:
Car Parking Fee for 3 hours: $30
```

📝 Problem 8: Banking Loan System

📌 Problem Statement:

Design a **Bank Loan System** where different types of loans are provided:

1. **Home Loan**
2. **Car Loan**
3. **Personal Loan**

Each loan type should have a function to calculate the **total repayment amount** based on interest rates and tenure.

🔷 **Hint:** Use an **Abstract Class** `Loan`.

🔻 **Example Input & Output:**

```
Input:
Loan* l1 = new HomeLoan();
l1→calculateRepayment(100000, 5);

Output:
Home Loan Repayment: $120000
```

📝 Problem 9: University Grading System

📌 **Problem Statement:**

Create a **University Grading System** where students are evaluated based on:

1. **Theory Exams**
2. **Practical Exams**
3. **Project Work**

Each evaluation type should have a function to calculate the **final grade** based on scores.

🔷 **Hint:** Use an **Abstract Class** `Evaluation`.

🔻 **Example Input & Output:**

```
Input:
Evaluation* e1 = new TheoryExam();
e1→calculateGrade(85);

Output:
Theory Exam Grade: A
```

📝 Problem 10: E-Commerce Discount System

📌 **Problem Statement:**

Implement an **E-Commerce Discount System** where different types of discounts apply based on:

1. **Festive Discount**
2. **Loyalty Discount**
3. **Bulk Purchase Discount**

Each discount type should implement a method to **calculate final price** after applying the discount.

🔷 **Hint:** Use an **Abstract Class** `Discount`.

🔻 **Example Input & Output:**

```
Input:
Discount* d1 = new FestiveDiscount();
d1→applyDiscount(1000);

Output:
Final Price after Festive Discount: $800
```

Happy Coding!