Object-Oriented Programming

Lab 4
ENSIA 2021/2022

Exercise 1 (10 minutes, to solve without using the PCs/laptops)

[To be done in class]

Exercise 2 (30 minutes)

Create a `PurchasesAccount` class. Use a `static` data member to contain the `annualDiscountRate` for each of the customers. Each member of the class contains a `private` data member `purchasesBalance` indicating the amount the customer currently has on debt. Provide a `calculateMonthlyDiscount` member function that calculates the monthly discount by multiplying the balance by `annualDiscountRate` divided by 12; this discount should be subtracted from the `purchasesBalance`. Provide a `static` member function `modifyDiscountRate` that sets the static `annualDiscountRate` to a new value.

Write a driver program to test class `purchasesAccount`. Instantiate two different `purchasesAccount` objects `customer1` and `customer2` with balances of 200 000 DA and 300 000 DA, respectively. Set `annualDicountRate` to 2%. Then calculate the monthly discount, and print the new balances for each of the customers. Then set `annualDiscountRate` to 3% and calculate the next month's discount and print the new balances for each of the customers.

Exercise 3 (30 minutes)

Suppose you need to process course information. Each course has a name and a number of students who take the course. You should be able to add/drop a student to/from the course. You can use a class to model the courses as described below the problem statement.

A `Course` object can be created using the constructor `Course (string courseName, int capacity)` by passing a course name and the maximum number of students allowed. You can add a student to the course using the `addStudent(string name)` function, drop a student from the course using the `dropStudent(string name)` function, and return all the students for the course using the `getStudents()` function. Given the definition of the class `Course`, define the functions members then give a test program that creates two courses and adds students to them.

```
1  class Course {
2      public:
3          Course (cont string & courseName, int capacity);
4          ~Course();
5          string getCourseName() const;
```

```
6         void addStudent(const string & name);
7         void dropStudent(const string & name);
8         string* getStudents() const;
9         int getNumberOfStudents() const;
10
11    private:
12        string courseName;
13        string * students;
14        int numberOfStudents;
15        int capacity;
16 };
```