Semi Final Activity 1 Documentation

Instructor: Cheeky Rose Villamarzo
Author: James Leo C. Grimaldo
Github Repository: C++_OOP

Files and folder

This activity consist of 2 program files main.cpp and interface.h, it can be found on github repository using this link Semi Final Activity 1

File Name	Description
main.cpp	Contains main method of the program.
interface.h	contains base class and 3 derived classes.
DOCUMENTATION.md	Contains documentation of this program

Class and Variable names

- Based on the given requirement, this program consist of variables that are used in main method and classes that act as a based class and derived classes.
  - `Installment` Class will act as based class consist of public varibles `tuitionFee`, `numUnits`, `amtPerUnit`, and `totalFee`. These variable will act as interface and will be used once the base class was inherited in derived classes. There are also varibles with pre-assigned values.

Variables	Data Type	Value	Description
`tuitionFee`	`double`	Null	Store computed amount using `computeTuitionFee()` method.
`numUnits`	`double`	Null	Store input numbers of units to be taken.
`amtPerUnit`	`double`	Null	Store input amouunt per units to be taken.
`totalFee`	`double`	Null	Store computed total fee using computeTotalFee() method.
`miscFee`	`double`	3000.00	Store Miscellaneous Fee
`labFee`	`double`	1000.00	Store Laboratory Fee

  - Derived classes are almost the same except its class names and the algorithm inside methods.

Derived Classes	Description

Derived Classes	Description
`class ThreeMonths`	consist of method that will compute installment fee for Three months.
`class FourMonths`	consist of method that will compute installment fee for four months.
`class FiveMonths`	consist of method that will compute installment fee for five months.

These derived classes consist of same method names that will compute Intsallment fee which is `displayInstallment()` that consist of one variable `installment` that will be returned once `displayInstallment` was called.

Variable Name	Data Type	Value	Description
`installment`	`double`	computeTotalFee() / 3;	this variable was used in `displayInstallment()` method under `ThreeMonths` class.
`installment`	`double`	computeTotalFee() / 4;	this variable was used in `displayInstallment()` method under `FourMonths` class.
`installment`	`double`	computeTotalFee() / 4;	this variable was used in `displayInstallment()` method under `FiveMonths` class.

Classes and Variables used in MAIN method

- Interface header file was imported in main file using `#include "interface.h"` in order to uses its classes.
Assigned Variable Name	Class Name	Class Type
`installment`	`Installment`	Base Class
`threeMonths`	`ThreeMonths`	Derived Class
`fourMonths`	`FourMonths`	Derived Class
`fiveMonths`	`FiveMonths`	Derived Class
Variable Name	Data Type	Value	Description
`numUnits`	`double`	Null	It will temporarily store input value of number of units.
`amtPerUnit`	`double`	Null	It will temporarily store input value of amount per Units.

Instructions:

- Using OOP Interface approach, write a program that will determine the monthly payment of school fee based on a installment basis.
    - Requirements:
        - ☑ Create a based class for installment fee. (10 pts)

```
//based class for installment fee
class Installment{
    public:
        double tuitionFee, numUnits, amtPerUnit, totalFee;
        double miscFee = 3000.00;
        double labFee = 1000.00;

        void setNumAndAmtperUnit(double numberOfUnits, double
amountPerUnit){
            numUnits   = numberOfUnits;
            amtPerUnit = amountPerUnit;
        }

        double computeTuitionFee(){
            tuitionFee = numUnits * amtPerUnit;
            return tuitionFee;
        }

        double computeTotalFee(){
            totalFee = computeTuitionFee() + miscFee + labFee;
            return totalFee;
        }

};
```

        - ☑ Create derived classes for 3 months, 4 months, and 5 months installment options.

```
//derived classes for 3 months
class ThreeMonths: public Installment{
    public:
        double installment;

        double displayInstallment(){
            installment = computeTotalFee() / 3;
            return installment;
        }
};
```

```
//derived classes for 4 months
class FourMonths: public Installment{
    public:
```

```
        double installment;


        double displayInstallment(){
            installment = computeTotalFee() / 4;
            return installment;
        }
};
```

```
//derived classes for 5 months
class FiveMonths: public Installment{
    public:
        double installment;


        double displayInstallment(){
            installment = computeTotalFee() / 5;
            return installment;
        }
};
```

- ☑ Assign a value to miscellaneous and laboratory fee. (6 pts).

```
double miscFee = 3000.00;
double labFee = 1000.00;
```

- ☑ Allow the user to input amount for number of units enrolled and amount per unit. (10 pts).

```
double numUnits, amtPerUnit;

cout<< "Enter number of units: ";
cin >> numUnits;
cout<< "Enter amount per unit: ";
cin >> amtPerUnit;
```

- ☑ Compute for tuition fee. Use the formula tuition fee = amount per unit x number of units enrolled. (5 pts)

```
double computeTuitionFee(){
    tuitionFee = numUnits * amtPerUnit;
    return tuitionFee;
}
```

- ☑ Compute for total fee. Use the formula total fee = tuition fee + misc fee + lab fee. (5pts)

```
double computeTotalFee(){
    totalFee = computeTuitionFee() + miscFee + labFee;
    return totalFee;
}
```

- ☑ Compute for monthly installment. Use the formula total fee / number of months. (5pts)

```
//method for 3 months.
double displayInstallment(){
    installment = computeTotalFee() / 3;
    return installment;
}
```

```
//method for 4 months.
double displayInstallment(){
    installment = computeTotalFee() / 4;
    return installment;
}
```

```
//method for 5 months.
double displayInstallment(){
    installment = computeTotalFee() / 5;
    return installment;
}
```

- ☑ Display all the necessary information such as:
  - ☑ tuition fee (5 pts)

```
cout<< "Tuition Fee: "<< installment.computeTuitionFee()
<<endl;
```

  - ☑ miscellaneous fee (2 pts)

```
cout<< "MISCELLANEOUS FEE: "<< installment.miscFee<<endl;
```

- ☑ laboratory fee (2 pts)

```
cout<< "LABORATORY FEE: "<< installment.labFee<<endl;
```

- ☑ total fee (5 pts)

```
cout<< "TOTAL FEE: " << installment.computeTotalFee()<<endl;
```

- ☑ monthly payment for 3, 4, and 5 months (10pts)

```
cout<< "3 MONTHS: " << threeMonths.displayInstallment()
<<endl;
cout<< "4 MONTHS: " << fourMonths.displayInstallment()<<endl;
cout<< "5 MONTHS: " << fiveMonths.displayInstallment()<<endl;
```

- ☑ display all amount in two decimal places (5 pts)

```
cout<< setprecision(2);
cout<< fixed;
```