Object-Oriented Programming
Fall 2015

Project 4

Seon Joo Kim

Outline

- Project Overview
- Programming Problems
- Requirements & Assumptions
- Deliverables, due-date and submission

Programming Problems 1

- Problem 1

  Write a class management system

- It’s a simple program to manage classes

- Class to implement
  - For classes
  - For students

Programming Problems 1

- Class student
  - Member variables for
    - student name
    - ID
  - Interface functions (you choose the names)

- Class class
  - Member variables for
    - Class name
    - Instructor
    - Student list
      - An array pointer to store students of the class

Programming Problems 1

- Class class
  - Member functions
    - Withdraw
      - Input : student ID
      - Delete student from student list of the class
      - Must use **delete** operator
    - Register
      - Input : student's name and student's ID
      - Create new student corresponding name and ID
      - Must use **new** operator
    - Display
      - Print information about the class
        - Name, instructor, number of registered student

** Extra Credit for overloading << to handle cout that does the same thing as the Display function

Outline

- Project Overview
- Programming Problems
- Requirements & Assumptions
- Deliverables, due-date and submission

First, Make a Class OOP with a class name and a professor name

Register 3 students and display all students in OOP class

Make a class ‘Data structure’ from a OOP class using ‘operater =‘ and change a class name and a professor name

Make a class ‘Network’ from a OOP class using ‘copy constructor’

```
int main()
{

    Class OOP("OOP", "Kim");

    OOP.Register("Aisha", 2015123123);
    OOP.Register("Smith", 2015123124);
    OOP.Register("Jenny", 2015123125);
    OOP.Display();


    Class DATA=OOP;

    DATA.SetClassName("Data Structure", "Yang");
    DATA.Withdraw(2015123123);
    DATA.Display();


    OOP.Display();


    Class Network(OOP);

    Network.SetClassName("Network", "Han");
    Network.Register("Esmay", 2015123126);
    Network.Display();


    OOP.Display();

}
```

Withdraw student ‘Aisha’ using student number and display it

Register student ‘Esmay’ and display it

```
Professor :Kim       Class Name :OOP
================================================
Total number of students :3

Aisha     2015123123
Smith     2015123124
Jenny     2015123125


Professor :Yang       Class Name :Data Structure
================================================
Total number of students :2

Smith     2015123124
Jenny     2015123125


Professor :Kim       Class Name :OOP
================================================
Total number of students :3

Aisha     2015123123
Smith     2015123124
Jenny     2015123125


Professor :Han       Class Name :Network
================================================
Total number of students :4

Aisha     2015123123
Smith     2015123124
Jenny     2015123125
Esmay     2015123126


Professor :Kim       Class Name :OOP
================================================
Total number of students :3

Aisha     2015123123
Smith     2015123124
Jenny     2015123125


계속하려면 아무 키나 누르십시오 . . .
```

Withdraw student ‘Aisha’ using student number and display it

It does not change students member in OOP

Register student ‘Esmay’ and display it

It does not change students member in OOP

```
int main( )
{

    Class OOP("OOP", "Kim");

    OOP.Register("Aisha", 2015123123);
    OOP.Register("Smith", 2015123124);
    OOP.Register("Jenny", 2015123125);
    OOP.Display( );


    Class DATA=OOP;

    DATA.SetClassName("Data Structure", "Yang");
    DATA.Withdraw(2015123123);
    DATA.Display( );


    OOP.Display( );


    Class Network(OOP);

    Network.SetClassName("Network", "Han");
    Network.Register("Esmay", 2015123126);
    Network.Display( );


    OOP.Display( );

}
```

```
C:\Windows\system32\cmd.exe
Professor :Kim        Class Name :OOP
==============================================
Total number of students :3

Aisha      2015123123
Smith      2015123124
Jenny      2015123125


Professor :Yang        Class Name :Data Structure
==============================================
Total number of students :2

Smith      2015123124
Jenny      2015123125


Professor :Kim        Class Name :OOP
==============================================
Total number of students :3

Aisha      2015123123
Smith      2015123124
Jenny      2015123125


Professor :Han        Class Name :Network
==============================================
Total number of students :4

Aisha      2015123123
Smith      2015123124
Jenny      2015123125
Esmay      2015123126


Professor :Kim        Class Name :OOP
==============================================
Total number of students :3

Aisha      2015123123
Smith      2015123124
Jenny      2015123125


계속하려면 아무 키나 누르십시오 . . .
```

Outline

- Project Overview
- Programming Problems
- Requirements & Assumptions
- Deliverables, due-date and submission

Requirements & Assumptions

- Use dynamic memory allocation for the list of students (pointer array)
- The program must follow the **'Gang of Three Rule' (see lecture note Ch.10)**
  - Class **class** should have
    - Copy constructor
    - Member assignment operator
    - Destructor

  - All the Classes should follow the Data Hiding Principle

- Reasonable comments in English. (Important!)
- Reasonable Indentation. (Important!)

Outline

- Project Overview
- Programming Problems
- Requirements & Assumptions
- Deliverables, due-date and submission

Marking Criteria and Plagiarism

- **Marking Criteria**
  - Score is only given to programs that compile and produce the correct output.
  - Points are deducted for programs that produce compiler warnings. Hint: use the –Wall command-line parameter to eliminate all warnings.
  - Points deductions on programming style: provide comments in your code and use proper indentation of lines.
  - Please pay particular attention to the requested output format of your programs. Deviating from the requested output format results in points deductions.

- **Plagiarism (Cheating)**
  - All submissions are checked for plagiarism.
  - Once detected, no score will be given for the lab to all students involved in the plagiarism incident.

Deliverables

- This time, your are required to prepare following files for this project.
  - ✦ For programming the problem: project4.cpp

Warning: you will lose points if the file name is not proper!

Archiving the deliverables

- Please zip the all files of your submission to a single archive:

  $ tar –jcvf **project4_<student_id>.tbz2** project4.cpp

- To make sure, you can extract the archive file with following command:

  $ tar –jxvf **project4_<student_id>.tbz2**

- Please note that in the above command, all must be typed in on a single line!
  - The shell will wrap-around the text for you

Submitting your archive

- You are asked to upload your archive on YSCEC.
  - **project4_<student_id>.tbz2**
- Due date: Nov 4, 2015. 11:55pm
- For instructions on how to upload a file on YSCEC, please see Lecture Note 2

Outline

- Project Overview
- Programming Problems
- Requirements & Assumptions
- Deliverables, due-date and submission