Object-Oriented Programming
Fall 2015

Project 3

Seon Joo Kim

Outline

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

Project Overview

- Problem 1
  - Write Overloading operators (1)
    - Should write a program using the Operator Overloading
    - Simple program that compares ‘my amount’ to ‘others amount of money’
    - Make overloading operators: +, -, ==, -(negative), <<, >>
- Problem 2
  - Write Overloading operators (2)
    - Should write a program using the Operator Overloading
    - Make overloading operators: ++
    - Distinguish postfix from prefix

Project Overview

- Problem 3
  - Write Overloading operators (3)
    - Should write a program using the Operator Overloading
    - Make overloading operators: []
- Problem 4
  - Write a program that chooses a suitor
    - You should use the class 'Vector' in your program.
    - The class 'Vector' has member function 'erase' and 'begin'.

All problems are from your textbook. Please refer to your textbook (Ch.7,8)

for more details about the problems.

Outline

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

Programming Problems 1

- It's a simple program to practice for operators overloading.
- Should make the operators +, -, ==, -(negative), <<, >>
- Write a program that will execute as follows using the Money class.
- Just enter amounts of money in U.S. currency.
- Should use a 'friend functions' in Class.

```
Enter an amount of money: $13.31
Your amount is $13.31
My amount is $10.09
One of us is richer.
$13.31 + $10.09 equals $23.40
$13.31 - $10.09 equals $3.22
```

```
Enter an amount of money: $4.20
Your amount is $4.20
My amount is $10.09
One of us is richer.
$4.20 + $10.09 equals $14.29
$4.20 - $10.09 equals $-5.89
```

Programming Problems 2

- Should make the operators ++ both prefix and postfix.
- You should use the class ‘IntPair’ in your program.
- It’s very simple program.
- You should optionally assign a input value in main function. (do not using a ‘cin’)

```
Postfix a++: Start value of object a: 10 29
Value returned: 10 29
Changed object: 11 30
Prefix ++a: Start value of object a: 13 31
Value returned: 14 32
Changed object: 14 32
```

Programming Problems 3

- Should make the operators [].
- You should use the class ‘CharPair’ in your program.
- It’s very simple program.
- You assign a input value in main function.
  → a[1] = A, a[2] = B
- Change two letters when user entered and Display a input letters.

```
a[1] and a[2] are:
AB
Enter two letters (no spaces):
AM
You entered:
AM
```

Outline

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

Requirements & Assumptions

- Use Class for all programming problems.
- Use ‘const’ and ‘& (reference)’ whenever necessary.
- User Inputs
  - for problem 1,
    Input form are always ‘$’+’dollarsPart’+’centsPart’
    ex) Enter an amount of money: $13.31
    → dollarsPart: 13, centsPart: 31
  - for problem 2, Do not use a ‘cin’.
    → must optionally assign in main function.
  - for problem 3,
    Initial letters ‘AB’ assigned in main function.
- 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 problem1: project3_1.cpp
  - For programming problem2: project3_2.cpp
  - For programming problem3: project3_3.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 **project3_<student_id>.tbz2** project3_1.cpp project3_2.cpp project3_3.cpp

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

  $ tar –jxvf **project3_<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.
  - **project3_<student_id>.tbz2**
- Due date: Oct. 15, 2014 (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 ✓