Object-Oriented Programming

Fall 2015

Project 2

Seon Joo Kim

Outline

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

Project Overview

- Program 1
  - Program to demonstrate a very simple example of a class
    - should write a program using the class
    - simple program that checks birthday date
- Problem 2
  - Making a airline seat reservation system
    - Based on the understanding of what you have learned so far (array, class, etc.), you will write a simple object-oriented program.

Outline

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

Programming Problems 1

- It's a simple program for you to practice writing a Class.
- You should use the class 'DayOfYear' in your program.

```
class DayOfYear
{
public:
    int month;
    int day;
    void output( );
    // method 'output' prints the date
    // that this class contains
};
```

Programming Problems 1

- Write a program that will execute as follows using the DayofYear class.

```
Enter today's date:
Enter month as a number: 10
Enter the day of the month: 1
Enter your birthday:
Enter month as a number: 10
Enter the day of the month: 1
Today's date is October 1
Your birthday is October 1
Happy Birthday!
```

```
Enter today's date:
Enter month as a number: 10
Enter the day of the month: 1
Enter your birthday:
Enter month as a number: 4
Enter the day of the month: 20
Today's date is October 1
Your birthday is April 20
Happy Unbirthday!
```

output of the function ‘output’ in the DayofYear Class

Programming Problems 2

- Write a simple airline seat reservation system.
- We have 3 airlines: A, B, C
- Each Airline has 25 seats. (5 rows * 5 columns)
  - 5 seats for the first class and 20 seats for the economy class.
  - You must use 2 dimensional array member variable to represent this.
  - Every seat is specified by its x, y position.

ex)

```
  Airline A
  1 2 3 4 5
1 X O X X X
  ---------
2 O O O O O
3 O O X O O
4 O O O O O
5 O O O O O
```

O : available seat
X : reserved seat

- You can only contain 25 seats in one Class instance.
  - It means, one Class instance can only represent one airline.

Programming Problems 2

- Our system should provide these features:

1. Creating new account
2. Show Seat Status (availability map and number of remaining seats)
2. Login and Reserve a seat (up to 5) (by taking x, y coordinates of seat)
3. Login and Return a seat (by taking one of user's reserved seat)
4. Login and Show User's Reservation Condition

5. Error handling
    1) When user selects menu which is non-existing.
    2) When user selects reservation menu in the absence of available seats.
    3) When user tries to reserve a seat which is already reserved.
    4) When user selects return menu if user has no reservation.
    5) When user tries to return a seat which is not reserved.
    6) When user types invalid x or y coordinates. (out of scope: x,y > 5 or x,y < 1)
    7) When user ID is not existing or Password is wrong
    8) When user reserved seats above 5
    9) When the first class seats are full

Following example from next pages will help your understanding.

Programming Problems 2

- Example of execution. (For reference. You don't have to exactly follow this.)

```
####################################
This is Project#3-3 Solution
####################################

-----------------------------------------
-------- Airline Reservation System  --------
-------------    Menu List    ---------------

 1. Sign up
 2. Show status
 3. Reservation
 4. Reservation condition
 5. Returning
 0. Exit
Select-> 1

Input your ID(0-to menu) :OOP
Input your password : 1234
Hello OOP

-----------------------------------------
-------- Airline Reservation System  --------
-------------    Menu List    ---------------

 1. Sign up
 2. Show status
 3. Reservation
 4. Reservation condition
 5. Returning
 0. Exit
Select-> 2

# Remaining Seat: 75


  Airline A       Airline B       Airline C
  1 2 3 4 5       1 2 3 4 5       1 2 3 4 5
1 0 0 0 0 0     1 0 0 0 0 0     1 0 0 0 0 0
  ---------       ---------       ---------
2 0 0 0 0 0     2 0 0 0 0 0     2 0 0 0 0 0
3 0 0 0 0 0     3 0 0 0 0 0     3 0 0 0 0 0
4 0 0 0 0 0     4 0 0 0 0 0     4 0 0 0 0 0
5 0 0 0 0 0     5 0 0 0 0 0     5 0 0 0 0 0
```

←Feature no.1
Creating new ac count

←Feature no.2

Show seat statu s

(All seats are available)

Feature no.2→

The first row of seats i s for First class. Other s are for economy clas s

```
-----------------------------------------
-------- Airline Reservation System  --------
-------------    Menu List    ---------------

 1. Sign up
 2. Show status
 3. Reservation
 4. Reservation condition
 5. Returning
 0. Exit
Select-> 3

Input your ID (0-to menu):OOP
Input your password : 1234
Select Airline A, B, C (1-A, 2-B, 3-C 0-to menu)-> 1
How many seats do you need (up to 5) : 5
Input 1th X Y -> 1 1
Successfully reserved
Input 2th X Y -> 1 2
Successfully reserved
Input 3th X Y -> 1 3
Successfully reserved
Input 4th X Y -> 1 4
Successfully reserved
Input 5th X Y -> 1 5
Successfully reserved

-----------------------------------------
-------- Airline Reservation System  --------
-------------    Menu List    ---------------

 1. Sign up
 2. Show status
 3. Reservation
 4. Reservation condition
 5. Returning
 0. Exit
Select-> 3

Input your ID (0-to menu):OOP
Input your password : 1234
You cannnot reserve anymore
```

←Feature no.3

First login and Reserve a seat up to 5

(first row of Airline A ar e all reserved)

←Feature no.3

Cannot reserve more than 5 sea ts

Programming Problems 2

```
-----------------------------------------
-------- Airline Reservation System  --------
------------    Menu List    ---------------

 1. Sign up
 2. Show status
 3. Reservation
 4. Reservation condition
 5. Returning
 0. Exit
Select-> 1

Input your ID(0-to menu) :OOP2
Input your password : 1234
Hello OOP2

-----------------------------------------
-------- Airline Reservation System  --------
------------    Menu List    ---------------

 1. Sign up
 2. Show status
 3. Reservation
 4. Reservation condition
 5. Returning
 0. Exit
Select-> 3

Input your ID (0-to menu):OOP2
Input your password : 1234
Select Airline A, B, C (1-A, 2-B, 3-C 0-to menu)-> 1
How many seats do you need (up to 5) : 2
The first class seats are full
Do you want to reserve economy class (Yes->1, No->2) ? 1
Input 1th X Y -> 3 3
Successfully reserved
The first class seats are full
Do you want to reserve economy class (Yes->1, No->2) ? 2
```

←Feature no.1
System should able to handle multiple accounts

←Feature no.3

When the first class seats are full, ask whether they want economy seats or not

```
-----------------------------------------
-------- Airline Reservation System  --------
------------    Menu List    ---------------

 1. Sign up
 2. Show status
 3. Reservation
 4. Reservation condition
 5. Returning
 0. Exit
Select-> 2

# Remaining Seat: 69


  Airline A       Airline B       Airline C
   1 2 3 4 5       1 2 3 4 5       1 2 3 4 5
1  X X X X X    1  0 0 0 0 0    1  0 0 0 0 0
   ---------       ---------       ---------
2  0 0 0 0 0    2  0 0 0 0 0    2  0 0 0 0 0
3  0 0 X 0 0    3  0 0 0 0 0    3  0 0 0 0 0
4  0 0 0 0 0    4  0 0 0 0 0    4  0 0 0 0 0
5  0 0 0 0 0    5  0 0 0 0 0    5  0 0 0 0 0

-----------------------------------------
-------- Airline Reservation System  --------
------------    Menu List    ---------------

 1. Sign up
 2. Show status
 3. Reservation
 4. Reservation condition
 5. Returning
 0. Exit
Select-> 4

Input your ID (0-to menu):OOP
Input your password : 1234
1: Airline A seat 1,1
2: Airline A seat 1,2
3: Airline A seat 1,3
4: Airline A seat 1,4
5: Airline A seat 1,5
```

←Feature no.2

Show the seats. The mark ‘X’ is reserved seat

←Feature no.4

Login and Show User’s reservation condition

Programming Problems 2

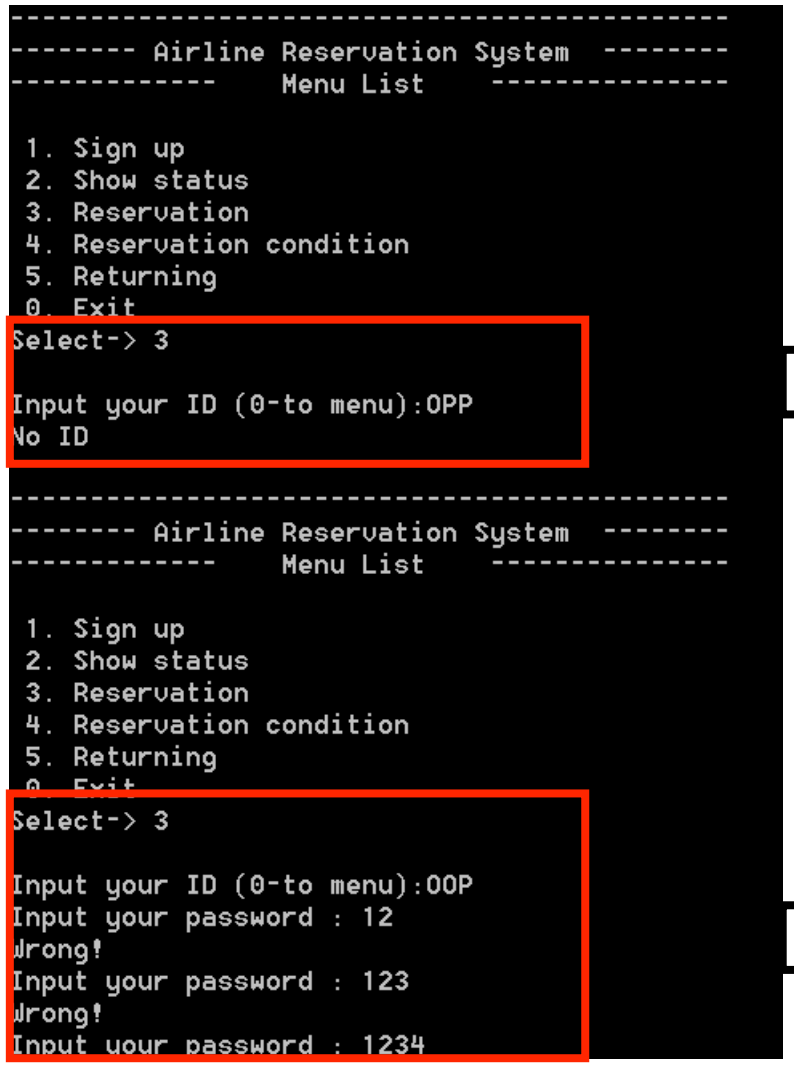
```
-----------------------------------------
-------- Airline Reservation System  --------
-------------    Menu List    ---------------

 1. Sign up
 2. Show status
 3. Reservation
 4. Reservation condition
 5. Returning
 0. Exit
Select-> 5

Input Your ID (0-to menu):OOP
Input your password : 1234
1: Airline A seat 1,1
2: Airline A seat 1,2
3: Airline A seat 1,3
4: Airline A seat 1,4
5: Airline A seat 1,5
Choose number :2
Successfully Returned

-----------------------------------------
-------- Airline Reservation System  --------
-------------    Menu List    ---------------

 1. Sign up
 2. Show status
 3. Reservation
 4. Reservation condition
 5. Returning
 0. Exit
Select-> 2

# Remaining Seat: 70


  Airline A       Airline B       Airline C
   1 2 3 4 5       1 2 3 4 5       1 2 3 4 5
1  X 0 X X X    1  0 0 0 0 0    1  0 0 0 0 0
   ---------       ---------       ---------
2  0 0 0 0 0    2  0 0 0 0 0    2  0 0 0 0 0
3  0 0 X 0 0    3  0 0 0 0 0    3  0 0 0 0 0
4  0 0 0 0 0    4  0 0 0 0 0    4  0 0 0 0 0
5  0 0 0 0 0    5  0 0 0 0 0    5  0 0 0 0 0
```

←Feature no.5 Show reservation condition and return selected seat.

```
-----------------------------------------
-------- Airline Reservation System  --------
-------------    Menu List    ---------------

 1. Sign up
 2. Show status
 3. Reservation
 4. Reservation condition
 5. Returning
 0. Exit
Select-> 7

Error (Please check Menu List Again)

-----------------------------------------
-------- Airline Reservation System  --------
-------------    Menu List    ---------------

 1. Sign up
 2. Show status
 3. Reservation
 4. Reservation condition
 5. Returning
 0. Exit
Select->
```

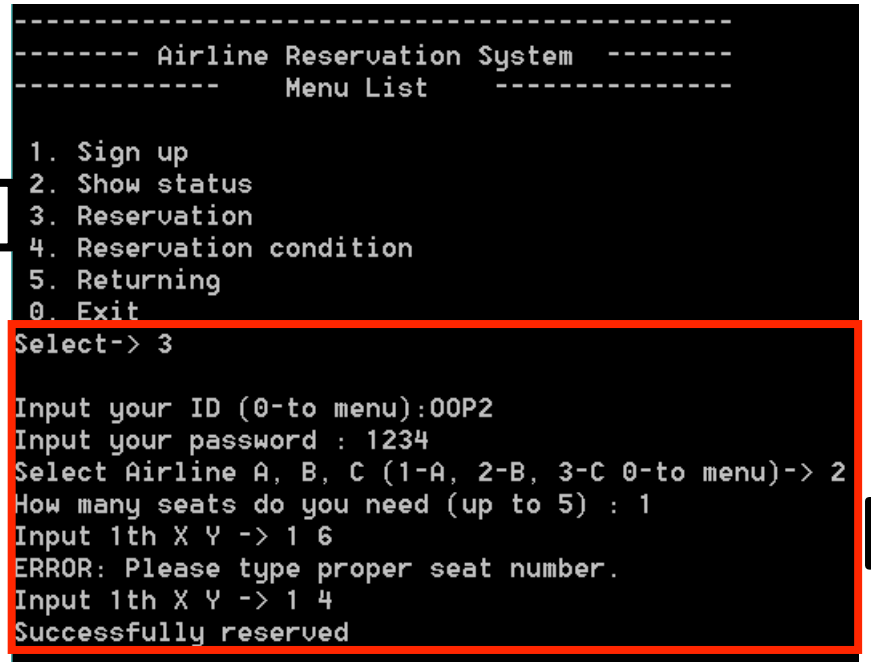
←Error handling

Programming Problems 2



←Error handling

←Error handling



←Error handling

Outline

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

Requirements & Assumptions

- Use Class for all programming problems.
- User Inputs(For problem 2)
  - Inputs are always unsigned integer for seat reservation

    (Position of seats)
  - However, you must check whether it's valid or not.
    (Requried feature no.4 'Error handling')
- 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 p roper indentation of lines.
  - Please pay particular attention to the requested output format of your programs. De viating 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 plagi arism incident.

Deliverables

- This time, your are required to prepare following files for this project.
  - For programming problem1: project2_1.cpp
  - For programming problem2: project2_2.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 project2_1.cpp project2_2.cpp proj
ect2_3.cpp

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

$ tar –jxvf project3_<student_id>.tbz2

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

Submitting your archive

- You are asked to upload your archive on YSCEC.
  - project2_<student_id>.tbz2

- Due date: **Oct. 12 (Mon), 11:55PM**