C COMPUTER PROGRAMMING (II) (1072) : Midterm Exam II

Cheng-Wen Ko.Ph.D, Associate Professor
E-Mail: cwko@cse.nsysu.edu.tw

Department of Computer Science and Engineering
National Sun Yat-sen University

May 27. 2019, 18:00:~ 21:00

General instructions:

1. Do not open this exam until you are told to begin.
2. Please turn off all cell phones and remove all headphones.
   手機請關機，並禁止使用任何耳機。
3. No USB disk or any portable device can be used.
   禁止使用任何可攜式電腦裝置以及USB 隨身碟。
4. You can only use the text editor for Ubuntu to edit your programs. You will lose 40 points for each time you are caught to use editors/IDEs for windows.
   你只能使用Ubuntu內建之文字編輯器進行程式編輯。每一次被發現使用Windows之 IDE 或文字編輯程式將扣此次考試成績40分。
5. This supplement has 10 pages including this cover.
6. You don’t need to provide the “makefile”, yet you are encouraged to do so. Alternatively, you can compile your code with these commands:
   ```
   g++ -c Q1.cpp
   g++ -o Q1.out Q1.o
   ```
7. Compress your programs (*.cpp and *.out) for all of the questions and name the compressed file with your student ID (example: B053040001.zip). Be sure that this file is transferred successfully to TA before you leave.
   作答完成後請將所有程式檔（*.cpp 以及 *.out ）壓縮成一個檔案，並且以學號命名。學號第一個字母請大寫。離開前請務必跟助教確認檔案已傳輸成功。

Q1.cpp

```
#include <iostream>
#include <string>
#include <vector>
using namespace std;

/* Please complete the program here by class definition
                                  and function definitions*/
// --------------------------------
// ----- ENTER YOUR CODE HERE -----
// --------------------------------


// --------------------------------
// --------- END USER CODE --------
// --------------------------------

// You cannot change this part of code
int main()
{
  char option;
  vector<Player> players;
  int numPlayers=0;

  do
  {
   cout << "Enter an option." << endl;
   cout << "a. Add new player and score." << endl;
   cout << "b. Print all players and scores." << endl;
   cout << "c. Search for a player's score." << endl;
   cout << "d. Remove a player." << endl;
   cout << "e. Quit." << endl;

   cin >> option;
   cout << endl;
   if (option == 'a')
     addPlayer(numPlayers, players);
   else if (option == 'b')
     printPlayer(numPlayers, players);
   else if (option == 'c')
     searchPlayer(numPlayers, players);
   else if (option == 'd')
     removePlayer(numPlayers, players);
   cout << endl;
  } while (option != 'e');

  return 0;
}
```

Example of Q1.out :

```
cse@pc100:~/Downloads$ ./Q1.out
Enter an option.
a. Add new player and score.
b. Print all players and scores.
c. Search for a player's score.
d. Remove a player.
e. Quit.
a

Enter new player name.
Allen
Enter new player score.
10

Enter an option.
a. Add new player and score.
b. Print all players and scores.
c. Search for a player's score.
d. Remove a player.
e. Quit.
a

Enter new player name.
Bob
Enter new player score.
20

Enter an option.
a. Add new player and score.
b. Print all players and scores.
c. Search for a player's score.
d. Remove a player.
e. Quit.
a

Enter new player name.
Ciro
Enter new player score.
30

Enter an option.
a. Add new player and score.
b. Print all players and scores.
c. Search for a player's score.
d. Remove a player.
e. Quit.
a

Enter new player name.
David
Enter new player score.
40

Enter an option.
a. Add new player and score.
b. Print all players and scores.
c. Search for a player's score.
d. Remove a player.
e. Quit.
a

Enter new player name.
Eddie
Enter new player score.
50

Enter an option.
a. Add new player and score.
b. Print all players and scores.
c. Search for a player's score.
d. Remove a player.
e. Quit.
a

Player array full.

Enter an option.
a. Add new player and score.
b. Print all players and scores.
c. Search for a player's score.
d. Remove a player.
e. Quit.
b

Player Scores
Allen 10
Bob 20
Ciro 30
David 40
Eddie 50

Enter an option.
a. Add new player and score.
b. Print all players and scores.
c. Search for a player's score.
d. Remove a player.
e. Quit.
c

What player to search for?
```

```
Bob
The score for Bob is 20

Enter an option.
a. Add new player and score.
b. Print all players and scores.
c. Search for a player's score.
d. Remove a player.
e. Quit.
c

What player to search for?
Frank
Player Frank not found.

Enter an option.
a. Add new player and score.
b. Print all players and scores.
c. Search for a player's score.
d. Remove a player.
e. Quit.
d

What player to remove?
Bob
 Player Bob removed.

Enter an option.
a. Add new player and score.
b. Print all players and scores.
c. Search for a player's score.
d. Remove a player.
e. Quit.
b

Player Scores
Allen 10
Eddie 50
Ciro 30
David 40

Enter an option.
a. Add new player and score.
b. Print all players and scores.
c. Search for a player's score.
d. Remove a player.
e. Quit.
a

Enter new player name.
Frank
```

```
Enter new player score.
60

Enter an option.
a. Add new player and score.
b. Print all players and scores.
c. Search for a player's score.
d. Remove a player.
e. Quit.
b

Player Scores
Allen 10
Eddie 50
Ciro 30
David 40
Frank 60

Enter an option.
a. Add new player and score.
b. Print all players and scores.
c. Search for a player's score.
d. Remove a player.
e. Quit.
e
```

Q2.cpp

```
#include <iostream>
#include <string>

using namespace std;

// --------------------------------
// ----- ENTER YOUR CODE HERE -----
// --------------------------------


// --------------------------------
// --------- END USER CODE --------
// --------------------------------


// ============================================================
//     main function -- You are not allowed to change here!
// ============================================================
int main()
{
  // Test our code with two Subscriber classes
  Subscriber s1, s2;
  //Test of default constructor
  cout << "Test of default constructor:" << endl;
  s1.OutputData();

  s1.InputData();        // Input data for Subscriber 1
  cout << "Subscriber 1's data:" << endl;
  s1.OutputData();       // Output data for Subscriber 1

  cout << endl;

  s2 = s1;
  cout << "Subscriber 2's data after assignment from Subscriber 1:"
     << endl;
  s2.OutputData();  // Should output same data as for Subscriber 1

  s1.ResetChannels();
  cout << "Subscriber 1's data after reset:" << endl;
  s1.OutputData();       // Should have no channels

  cout << "Subscriber 2's data, should still have original channels:"
     << endl;
  s2.OutputData();       // Should still have original channels

  cout << endl;
  return 0;
}
```

Example of Q2.out

```
cse@pc100:~/Downloads$ ./Q2.out
Test of default constructor:
Name:
Number of Channels: 0

Enter Subscriber name.
Ebba
Enter number of channels.
3
Enter name of channel 1
BBC
Enter name of channel 2
CNN
Enter name of channel 3
Netflix

Subscriber 1's data:
Name: Ebba
Number of Channels: 3
  Channel 1:BBC
  Channel 2:CNN
  Channel 3:Netflix


Subscriber 2's data after assignment from Subscriber 1:
Name: Ebba
Number of Channels: 3
  Channel 1:BBC
  Channel 2:CNN
  Channel 3:Netflix

Subscriber 1's data after reset:
Name: Ebba
Number of Channels: 0

Subscriber 2's data, should still have original channels:
Name: Ebba
Number of Channels: 3
  Channel 1:BBC
  Channel 2:CNN
  Channel 3:Netflix


cse@pc100:~/Downloads$
```

Q3.cpp

```
#include <iostream>
#include <string>
#include <vector>
using namespace std;

// --------------------------------
// ----- ENTER YOUR CODE HERE -----
// --------------------------------




// --------------------------------
// --------- END USER CODE --------
// --------------------------------


// ============================================================
//     main function -- You are not allowed to change here!
// ============================================================

int main()
{
     BoxOfProduce b1;
     b1.addBundle("Tomato");
     b1.addBundle("Potato");

     BoxOfProduce b2;
     b2.addBundle("Apple");
     b2.addBundle("Pear");
     b2.addBundle("Kiwi");
     b2.addBundle("Durian");

     BoxOfProduce b3;
     b3 = b1 + b2;

     cout << "box-1 " <<b1.boxContents() << endl;
     cout << "box-2 " <<b2.boxContents() << endl;
     cout << "box-3 " <<b3.boxContents() << endl;

     return 0;
}
```

Example of Q3.out

```
cse@pc100:~/Downloads$ ./Q3.out
box-1 The box contains:  (1)Tomato (2)Potato
box-2 The box contains:  (1)Apple (2)Pear (3)Kiwi (4)Durian
box-3 The box contains:  (1)Tomato (2)Potato (3)Apple (4)Pear (5)Kiwi
(6)Durian
cse@pc100:~/Downloads$
```

Q4.cpp

```
#include <string>
#include <iostream>
using namespace std;

// --------------------------------
// ----- ENTER YOUR CODE HERE -----
// --------------------------------


// --------------------------------
// --------- END USER CODE --------
// --------------------------------
// ============================================================
//     main function -- You are not allowed to change here!
// ============================================================
int main()
{
     DynamicStringArray names;

     // List of names
     names.addEntry("Allen");
     names.addEntry("Bob");
     names.addEntry("Ciro");
     names.addEntry("David");
     names.addEntry("Eddie");

     // Output list
     cout << "List of names:" << endl;
     for (int i = 0; i < names.getSize(); i++)
          cout << names.getEntry(i) << endl;
     cout << endl;

     // Add and remove some names
     names.addEntry("Frank");
     cout << "After adding a name:" << endl;
     for (int i = 0; i < names.getSize(); i++)
          cout << names.getEntry(i) << endl;
     cout << endl;
     names.deleteEntry("Bob");
     cout << "After removing a name:" << endl;
     for (int i = 0; i < names.getSize(); i++)
          cout << names.getEntry(i) << endl;
     cout << endl;

     names.deleteEntry("Garfield");
     cout << "After removing a name that isn't on the list
(Garfield):” << endl;
     for (int i = 0; i < names.getSize(); i++)
          cout << names.getEntry(i) << endl;
     cout << endl;

     names.addEntry("Hamlet");
     cout << "After adding another name:" << endl;
     for (int i = 0; i < names.getSize(); i++)
          cout << names.getEntry(i) << endl;
     cout << endl;

     // Remove all of the names by repeatedly deleting the last one
     while (names.getSize() > 0) {
        names.deleteEntry(names.getEntry(names.getSize() - 1));
     }

     cout << "After removing all of the names:" << endl;
     for (int i = 0; i < names.getSize(); i++)
          cout << names.getEntry(i) << endl;
     cout << endl;

     names.addEntry("Isaac");
     cout << "After adding a name:" << endl;
     for (int i = 0; i < names.getSize(); i++)
          cout << names.getEntry(i) << endl;
     cout << endl;

     cout << "Testing copy constructor" << endl;
     DynamicStringArray names2(names);
     names.deleteEntry(“Isaac"); // Remove Isaac from names
     cout << "Copied names:" << endl;
     for (int i = 0; i < names2.getSize(); i++)
          cout << names2.getEntry(i) << endl;
     cout << endl;

     cout << "Testing assignment" << endl;
     DynamicStringArray names3 = names2;
     // Remove Isaac from names2
     names2.deleteEntry("Isaac");
          cout << "Copied names:" << endl;
     for (int i = 0; i < names3.getSize(); i++)
          cout << names3.getEntry(i) << endl;
     cout << endl;
     cout << "Enter a character to exit." << endl;
     char wait;
     cin >> wait;
     return 0;
}
```

Example of Q4.out

```
cse@pc100:~/Downloads$ ./Q4.out
List of names:
Allen
Bob
Ciro
David
Eddie

After adding a name:
Allen
Bob
Ciro
David
Eddie
Frank

After removing a name:
Allen
Ciro
David
Eddie
Frank

After removing a name that isn't
on the list (Garfield):
Allen
Ciro
David
Eddie
Frank

After adding another name:
Allen
Ciro
David
Eddie
Frank
Hamlet

After removing all of the names:
```

```
After adding a name:
Isaac

Testing copy constructor
Copied names:
Isaac

Testing assignment
Copied names:
Isaac

Enter a character to exit.
k
cse@pc100:~/Downloads$
```

(End of the Exam)