

AST21105 Object-Oriented Programming & Design

Lab 5 – Pointers and Dynamic Allocation II

A. Submission Details

In this lab, you are required to submit **TWO** C++ programs to solve the given problems shown in the section “Exercise”. To make the program implementation easier, you are suggested to write your program using Visual Studio .NET instead of doing it directly using paper & pencil. After you have completed the implementation, submit your program files (i.e. NumberGuessing.cpp and MultiplyPolynomials.cpp for this lab) using the electronic “drop-box” in Canvas, **One week** after your lab section is conducted. For details, please refer to the following:

| | |
|---------------------------|---|
| <i>Tuesday sections</i> | <i>by 19:00 on 23 February 2016 (Tuesday)</i> |
| <i>Wednesday sections</i> | <i>by 19:00 on 24 February 2016 (Wednesday)</i> |

You are reminded to double-check your solution to verify everything is correct before submission. You are also required to put your name and student ID at the beginning of your source file.

Important: You are only able to submit your work once.

B. Objective

The objective of this lab is to help you get familiar with dynamic array allocation and its manipulation that you have learned in last two lectures. As pointers are highly related to arrays in C++, more training on arrays will be given so as to reinforce your concepts.

The first part of this lab is meant to be a simple review of pointer and array declaration and manipulation using Microsoft Visual Studio .NET. The last part of this lab is two practical tasks that allows you to use all the techniques introduced (i.e. pointers and arrays).

C. Review

In this lab, we demonstrate how to pass a dynamically allocated array to a function.

Examples:

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

// Passing array to function
void changeValVer1( int* const arr, int size ) {
    for(int i=0; i<size; i++)
        arr[i] = i;
}

void changeValVer2( int arr[], int size ) {
    for(int i=0; i<size; i++)
        arr[i] = i;
}

void displayElements( int* const arr, int size ) {
    for(int i=0; i<size; i++)
        cout << arr[i] << " "; cout << endl;
}
```

```

int main() {
    cout << "Enter size of array: ";
    int n;
    cin >> n;
    int* arr = new int[n];
    for(int i=0; i<n; i++)
        arr[i] = 0;

    changeValVer1(arr, n);
    displayElements(arr, n);

    for(int i=0; i<n; i++)
        arr[i] = 0;

    changeValVer2(arr, n);
    displayElements(arr, n);

    delete [] arr;
    system("pause");
    return 0;
}

```

Question:

Which function(s), changeValVer1, changeValVer2 does (do) value change of array correctly?

Answer: Both of them do the value change of array correctly.

D. Exercise

Task 1:

In this part, you are required to:

- i) Create a New Project and give your project the name Lab5a.
- ii) Add a source file to your project, called NumberGuessing.cpp
- iii) In this lab, you will implement a number guessing game. Your task is to crack a code which has been randomly created by the computer. To crack the code, you make guesses and the computer responds with clues.
 - At the beginning, player will be asked to input a number n (an integer value), which corresponds to the size of an array.
 - Then, an array of size n will be created and each array element will be initialized to a number in range 1 to n.
 - Next, the array will be shuffled with the following function:

```
// "randNoArr" is a constant pointer to a dynamically created array
// "size" is the size of the dynamic array pointed by randNoArr

void shuffle( int* const randNoArr, int size )
{
    for( int i=0; i<(size-1); i++ )
    {
        int r = i + (rand() % (size-i));
        int temp = randNoArr[i];
        randNoArr[i] = randNoArr[r];
        randNoArr[r] = temp;
    }
}
```

- Making Guesses
 - To make a guess of the number sequence, simply fill in a row with digits between 1 – n separated by a space.
- Using the clues
 - A “O” symbol means “Right value” and “Right position”.
 - A “X” symbol means either “Wrong value” or “Wrong position”

- Example output should look like this:

Enter total number: 6

Number Guessing

Enter 6 digits (1-6) separated by a space

Round 1

Enter Guess: 1 2 3 4 5 6

X X X X X X

Number Guessing

Enter 6 digits (1-6) separated by a space

Round 2

Enter Guess: 1 3 2 5 6 4

X X X X O X

Number Guessing

Enter 6 digits (1-6) separated by a space

Round 3

Enter Guess: 3 2 1 4 6 5

O X X X O X

Number Guessing

Enter 6 digits (1-6) separated by a space

Round 4

Enter Guess: 3 4 5 2 6 1

O O O X O X

Number Guessing

Enter 6 digits (1-6) separated by a space

Round 5

Enter Guess: 3 4 5 1 6 2

O O O O O O

Congratulations! You win in 5 steps

The input which is underlined is the user's input to a question.

- Another example output is shown below:

```
Enter total number: 5
Number Guessing
Enter 5 digits (1-5) separated by a space
-----
Round 1
Enter Guess: 1 2 3 4 5
              O O X X X
-----
Number Guessing
Enter 5 digits (1-5) separated by a space
-----
Round 2
Enter Guess: 1 2 4 3 5
              O O X O X
-----
Number Guessing
Enter 5 digits (1-5) separated by a space
-----
Round 3
Enter Guess: 1 2 5 3 4
              O O O O O
-----
Congratulations! You win in 3 steps
```

The input which is underlined is the user's input to a question.

- iv) Compile your program and test it by executing your program.

Important note:

You are required to **decompose your program into a number of functions** in order to obtain full marks for this lab. Also, you have to make sure there is no memory leakage in your program.

Task 2:

In this part, you are required to:

- i) Create a New Project and give your project the name Lab5b.
- ii) Add a source file to your project, called MultiplyPolynomials.cpp
- iii) Write a C++ program to implement a polynomial solver. Your code should perform the following:
 - A) Allow a user to input **TWO** polynomials with any number of terms according to user input. Each polynomial could be represented by a set of double values. For example:

Input: 5 0 -3.5 4 5 corresponds to polynomial: $p(x) = 5 - 3.5x^2 + 4x^3 + 5x^4$
Input: -2.8 0 6 corresponds to polynomial: $q(x) = -2.8 + 6x^2$
 - B) Implement a function **multiplyPolynomials** to multiply the two input polynomials together and return the result.
 - C) The output of the program should look like below:

Polynomial Solver

Number of terms for the first polynomial p: 5

Terms (from lowest order term to the highest): 5 4 -3.5 0 5

Number of terms for the second polynomial q: 3

Terms (from lowest order term to the highest): 6 0 -2.8

Result: 30, 24 x1, -35 x2, -11.2 x3, 39.8 x4, -14 x6

Number of terms for the first polynomial p: 3

Terms (from lowest order term to the highest): 1 2.5 3

Number of terms for the second polynomial q: 4

Terms (from lowest order term to the highest): 2 0 1.8 6.4

Result: 2, 5 x1, 7.8 x2, 10.9 x3, 21.4 x4, 19.2 x5

Number of terms for the first polynomial: -1

Thanks for using our program!

iv) Assuming the function prototype is as follows:

```
double* multiplyPolynomials ( const double* const p,
                             const double* const q,
                             int np, int nq);
```

v) Compile your program and test it by executing your program.

Important note:

You are required to **make sure there is no memory leakage** in your program.

Marking Scheme:

| Graded items | Weighting |
|--|-----------|
| 1. Correctness of program (i.e. whether your code is implemented in a way according to the requirements as specified.) | 60% |
| 2. Indentation | 30% |
| 3. Documentation (with reasonable amount of comments embedded in the code to enhance the readability.) | 10% |
| | 100% |

Program Submission Checklist

Before submitting your work, please check the following items to see you have done a decent job.

Items to be checked

☒ / ☒

1. Did I put my name and student ID at the beginning of all the source files? ☐
2. Did I put reasonable amount of comments to describe my program? ☐
3. Are they all in .cpp extension and named according to the specification? ☐
4. Have I checked that all the submitted code are compliable and run without any errors? ☐
5. Did I zip my source files using Winzip / zip provided by Microsoft Windows? Also, did I check the zip file and see if it could be opened?
(Only applicable if the work has to be submitted in zip format.) ☐
6. Did I submit my lab assignment to Canvas? ☐

-End-