National University of Computer and Emerging Sciences



Lab Manual 01

Object Oriented Programming

|  |  |
| --- | --- |
| Course Instructor | Miss Abeeda Akram |
| Lab Instructor (s) | Miss. Siddiqua Nayyer  Mr. Dilawar Shabbir |
| Section | BCS – 2A |
| Semester | Spring 2021 |

Department of Computer Science

FAST-NU, Lahore, Pakistan

## Objectives

After performing this lab, students shall be able to:

* Have an improved understanding of pointers.
* Access and modify pointers in functions.
* How pointers and array can be related.
* Debugging with pointers.

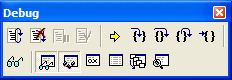
**TASK 1:**

Write the C++ code for a function swap(), which swaps the values of two integer variables.

Note: you cannot create global variables, and cannot pass integer variables by reference in the function.

Hint: Pointers

Debugging Commands of VS for help:

** **

|  |  |  |  |
| --- | --- | --- | --- |
| Short cut key | Icon | Menu | Explanation |
| F-9 |  |  | Insert/Remove breakpoint |
| F-5 |  | Debug-Go | Execute a program until the next breakpoint |
| Shift F-5 |  | Debug-Stop debugging | To stop debugging a program. It will stop executing the program |
| F-10 |  | Debug-StepOver | Go to the next statement |
| F-11 |  | Debug-Step Into | Go inside a function |
| Shift F-11 |  | Debug – Step Out | Come out of the function |
|  |  | Debug - Run to cursor | Execute all statements till the statement on which the cursor is placed or until the next breakpoint |
| Alt -3 |  | Debug-Windows-Watch | Show the window where only the variables in scope are shown |
| Alt-4 |  | Debug-Windows-Variables | Show the window in which you can type a variable name to see its value |
| Alt-7 |  | debug-windows-call stack | You can see the activation of stack of functions here |

**TASK 2:**

Given two integers x and y, find their sum using pointers.

**TASK 3:**

Run this code and check if there is any issue, also detect dangling pointer and memory leak.

int\*ptr = new int;

cout << "Enter Int Value: ";

cin >> \*ptr;

cout <<"Value is: " <<\*ptr << endl;

cout << "Pointer Value is: " << ptr << endl;

delete ptr;

cout << "After Del,,Value is: " << \*ptr << endl;

cout << "After Del,,Pointer Value is: " << ptr << endl;

cout << "Dangling Pointer ? If Yes, then Resolve issue" << endl;

cout << "Dynamically occupied Float Variable: "<<new float << endl;

int\*ptr1 = new int;

\*ptr1=9;

cout << \*ptr1 << endl;

ptr1++;

cout << \*ptr1 << endl;

cout << "Find Memory Leak" << endl;

**TASK 4:**

1. Introduce 2 variables i (int), j (float) and initialize to 5, 1.5 respectively.
2. Introduce 2 variables xPtr (int\*), yPtr (float\*) and assign address of i, j respectively.
3. Print values of i, j and addresses in xPtr, yPtr.
4. xPtr ++; yPtr ++;
5. Print addresses in xPtr,yPtr.
6. xPtr --; yPtr --;
7. Print addresses in xPtr, yPtr.
8. xPtr = xPtr + 3; yPtr = yPtr + 4;
9. Print addresses in xPtr, yPtr.

**TASK 5:**

What is wrong with the following code?

double \*firstPtr = new double; //Line 1

double \*nextPtr = new double; //Line 2

\*firstPtr = 62; //Line 3

nextPtr = firstPtr; //Line 4

delete firstPtr; //Line 5

delete nextPtr; //Line 6

firstPtr = new double; //Line 7

\*firstPtr = 28; //Line 8

cout << \*firstPtr << " " << \*nextPtr << endl; //Line 9

**TASK 6:**

Create two float pointers in the main function and write C++ code for the following functions and call them in order from main.

1- A function allocate(), which creates dynamic variables length and width and assigns their addresses to pointers created in main.

2- A function input() that takes input from the user in already created dynamic variables length and width.

3- A function Print() that prints values of dynamic variables length and width.

4- A function Area() that takes dynamic variables as parameters and returns the area of the rectangle.

5- A function deallocate() which deallocates the memory of dynamic variables length and width and update pointers too.