# Department of Computing

# School of Electrical Engineering and Computer Science

**CS-250: Data Structure and Algorithms**

**Class: BESE 13AB**

# 

# Lab 1: Revision of Pointers in C++

**Name:** Aimen Munawar

**CMS is:** 415867

**Class:** BESE-13-A

**Date: 15th September, 2023**

**Time: 9:00 am – 12:50 pm**

# Instructor: Dr. Faisal Shafait & Dr. Sidra

# Lab Engineer: Anum Asif

# Lab 1: Pointers in C++

**Introduction**

This lab is about the pointers. In C++, a pointer refers to a variable that holds the address of another variable. Like regular variables, pointers have a data type. For example, a pointer of type integer can hold the address of a variable of type integer. A pointer of character type can hold the address of a variable of character type.

**Objectives**

This lab will revise the old concepts taught to the students in the previous semesters.

**Tools/Software Requirement**

Visual Studio C++

**Description**

Pointers are used to point towards a particular memory address. In this lab we will use the pointers and perform task with the help of them.

**Lab Tasks**

**Task 1**

Write code to find the memory in bytes occupied by int, long, double, float and char.

**Code:**

#include<iostream>

using namespace std;

/\*Lab 1\*/

/\*Task 1 :Find the memory in bytes occupied by int, long, double, float and char.\*/

int main(){

int a;

long b;

double c;

float d;

char e;

cout << "Size of int:" << sizeof(a)<<endl;

cout << "Size of long:" << sizeof(b) << endl;

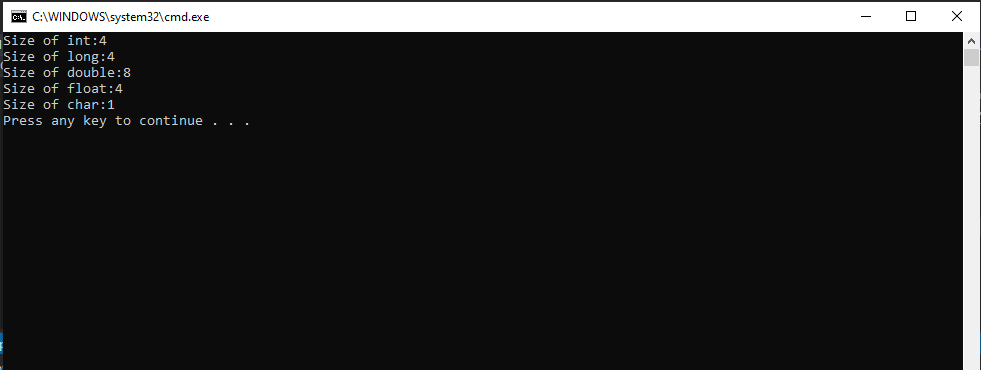
cout << "Size of double:" << sizeof(c) << endl;

cout << "Size of float:" << sizeof(d) << endl;

cout << "Size of char:" << sizeof(e) << endl;

return 0;

}



**Task 2**

Consider the following program and answer the questions.

void main()

{

int a, \*pa; // Statement 1

pa = &a; // Statement 2

cout<<"pa = &a --> pa = "<<pa<<endl<<endl;

pa = pa + 1; // Statement 3

cout<<"pa = pa + 1 --> pa = "<<pa<<endl<<endl;

pa = pa + 3; // Statement 4

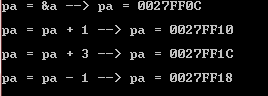
cout<<"pa = pa + 3 --> pa = "<<pa<<endl<<endl;

pa = pa - 1; // Statement 5

cout<<"pa = pa - 1 --> pa = "<<pa<<endl<<endl;

}

Output:



1. Why does the memory address stored in pointer “pa” vary by 4?

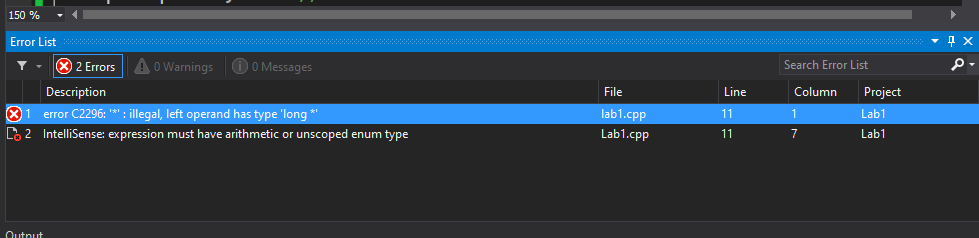
**Memory address stored in pointer “pa” vary by 4 because the pointer pa points to an integer that takes 4 bytes in memory .so ,whenever we add or subtract 1 from pa it will point to the object that is 4 bytes after/before the variable ‘a ‘ and hence the memory address stored in it will vary by ‘4’.**

1. Will the address still vary by 4 if the data type of the above mentioned code changed from “int” to “long”? Explain your answer.

**Yes, the address will still vary by 4 because both int and long datatype occupy 4 bytes in memory.**

1. If we try to multiply the address pointed to by “pa” what will happen? Is this logically or programmatically correct? Attach screen shot of the output you get when you try this multiplication.

**No, it is not correct programmatically because in C++ you can’t multiply a pointers value by a constant you can only add a pointers value in a constant.**



**Task 3**

int list[5]={3,6,9,12,15};

int \*pArr= list;

Your task is to write a piece of code that prints all values stored in the array **list** using only pointer variable pArr. Do not use the conventional way of printing values by numbering indexes.

Code:

#include<iostream>

using namespace std;

/\*Lab 1\*/

/\*Task 3\*/

int main()

{

int list[5] = { 3, 6, 9, 12, 15 }**;// creating an array of 5 elements**

int \*pArr = list; **// creating a pointer to that store array**

for (int i = 0; i < 5; i++){

**// loop to print values in array one after another because in pointers values are stored in continuous memory location**

cout << \*(pArr + i)<<endl;

**// printing the values stored in (pArr + i ) location**

}

return 0;

}



**Task 4**

Write output of the following C++ codes in your document without executing it.

**Example code a)**

int x[4] = {0,4,6,9};  
int \*p, a=3;  
p=x;  
(\*p)++;  
cout<<\*p<<endl;  
cout<<\*(p+1)<<endl;  
p++;  
\*p=\*p+a;  
cout<<\*p<<endl;  
p=p+2; //What is happening here?  
cout<<\*p<<endl;

**Output:**

**1**

**4**

**7**

**9**

**Example code b)**

int a, \*p, \*q;

int arr[4]= {0};

p=arr;

q=p;

\*p=4;

for(int i=0; i<3; i++){

a=\*p;

p++;

\*p=(a+i);

}

for (int j=0; j<4; j++){

cout<<\*q<<" ";

q++;

}

**Output:**

**4 4 5 7**

**Task 5**

int a=5, b=10;

int \*pa=&a; //pa and pb are pointer variables of type int.

int \*pb=&b;

int \*\*ppa=&pa; //ppa and ppb are called double pointers or pointers-to-pointers.

int \*\*ppb=&pb;

1. Write code of a function that swaps values of variables a and b. Input to the function should be the address of both the variables.

**Code:**

#include<iostream>

using namespace std;

/\*Lab 1\*/

/\*Task 5 : Part a\*/

void swap(int \*pa, int \*pb){

int temp;

temp=\*pa;

\*pa = \*pb;

\*pb = temp;

}

int main()

{

int a = 5, b = 10;

int \*pa = &a; //pa and pb are pointer variables of type int.

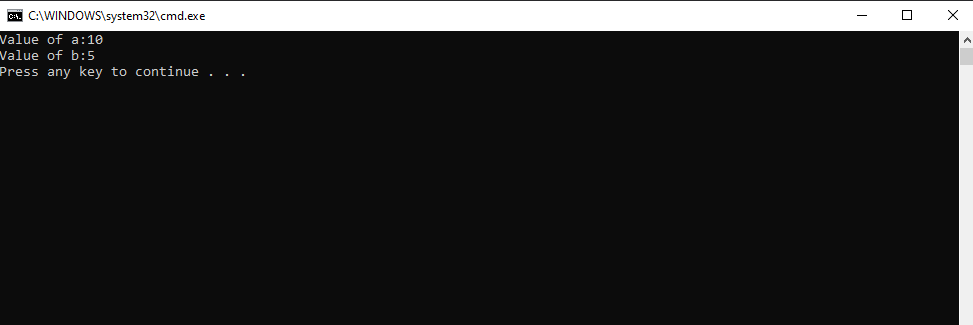
int \*pb = &b;

swap(&a, &b);

cout << "Value of a:" << a <<endl;

cout << "Value of b:" << b << endl;

}



1. Write code of a function that swaps values of the variables a and b using pointer-to-pointer variables ppa and ppb.

**Code:**

#include<iostream>

using namespace std;

/\*Lab 1\*/

/\*Task 5 : Part b\*/

void swap(int \*\*pa, int \*\*pb){

int temp;

temp=\*\*pa;

\*\*pa = \*\*pb;

\*\*pb = temp;

}

int main()

{

int a = 5, b = 10;

int \*pa = &a; //pa and pb are pointer variables of type int.

int \*pb = &b;

int \*\*ppa = &pa; //ppa and ppb are called double pointers or pointers-to-pointers.

int \*\*ppb = &pb;

swap(ppa, ppb);

cout << "Value of a:" << a <<endl;

cout << "Value of b:" << b << endl;

}



**Deliverables**

Compile a single word document by filling in the solution part and submit this Word file on LMS. The name of word document should follow this format. i.e. **YourFullName(reg)\_Lab#.** You must show the implementation of the tasks in a complete Word document to get your work graded. You must also submit this Word document on the LMS.

**Note:** Students are required to upload the lab on LMS before deadline.

Use proper indentation and comments. Lack of comments and indentation will result in deduction of marks.