

Lab 5: Pointer

Pointer is *a variable that stores memory addresses*. Unlike normal variables it does not store user given or processed value, instead it stores valid computer memory address.

- Pointers are more efficient in handling arrays and structures.
- Pointers are used to return multiple values from a function.
- Pointer allows dynamic memory allocation and deallocation (*creation and deletion of variables at runtime*) in C++. Which undoubtedly is the biggest advantage of pointers.
- Pointer allows to refer and pass a function as a parameter to functions

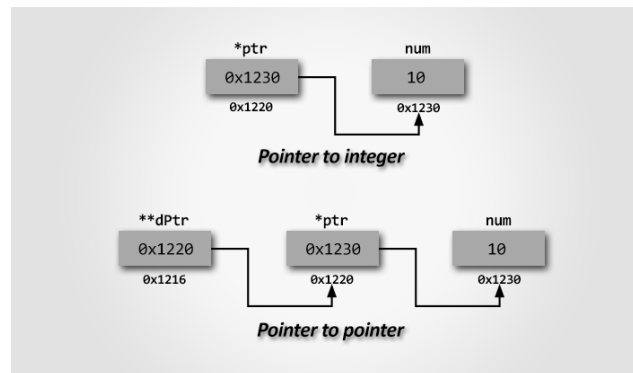


Figure 1: pointer

*	dereference operator, indirection operator	This is used to declare a variable as a pointer. It is also used when you want to access the value pointed to by the pointer variable.
&	reference operator, address-of operator	Use before a variable to indicate that you mean the address of that variable. You'll often see this in a function header where the parameter list is given.
->	member selection operator	This is used to refer to members of structures or class.

Part A: Demonstrate the use of pointers

1. Complete the below C++ program, so that it can read the memory address of any variable.

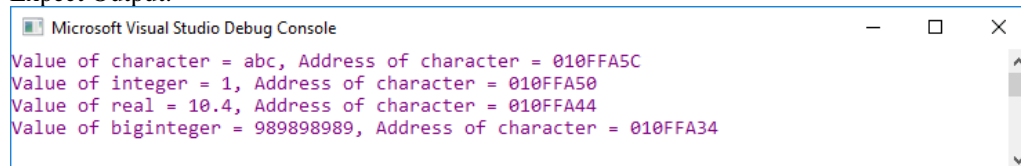
```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    /* Simple declarations */
    string word = "abc";
    int integer = 1;
    float real = 10.4f;
    long long biginteger = 98989898911;

    /* Print variable value with their memory address */
    //complete this section

    return 0;
}
```

Expect Output:



```
Microsoft Visual Studio Debug Console
Value of character = abc, Address of character = 010FFA5C
Value of integer = 1, Address of character = 010FFA50
Value of real = 10.4, Address of character = 010FFA44
Value of biginteger = 989898989, Address of character = 010FFA34
```

[Estimate Finish Time: 10 minutes]

2. Below is a C++ program to create, initialize and use pointer variable. Execute the program and find its final output.

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

int main()
{
    int num = 10;
    int * ptr;

    /* Stores the address of num to pointer type */
    ptr = &num;

    cout << "Address of num = " << &num << endl;
    cout << "Value of num = " << num << endl << endl;

    cout << "Address of ptr = " << &ptr << endl;
    cout << "Value of ptr = " << ptr << endl;
    cout << "Value pointed by ptr = " << *ptr << endl;

    return 0;
}
```

[Estimate Finish Time: 10 minutes]

Test yourself questions for Part A:

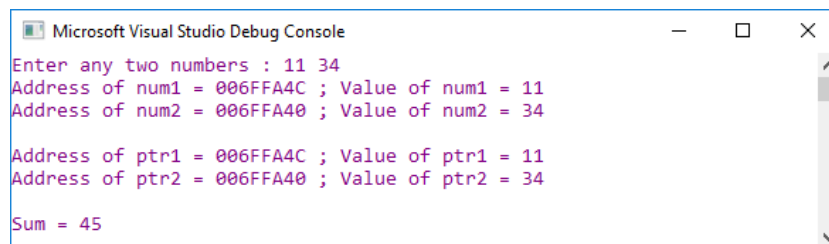
1. Write a C++ program to read two numbers from user and add them by using the pointers. Your program output should display the similar content as the output sample.

Given below variables and pointer variables:

```
// variable sum - store the answer from the pointers' summation operation.
int num1, num2, sum;

//use these pointers to do the summation operation.
int *ptr1, *ptr2;
```

Output sample:



```
Microsoft Visual Studio Debug Console
Enter any two numbers : 11 34
Address of num1 = 006FFA4C ; Value of num1 = 11
Address of num2 = 006FFA40 ; Value of num2 = 34

Address of ptr1 = 006FFA4C ; Value of ptr1 = 11
Address of ptr2 = 006FFA40 ; Value of ptr2 = 34

Sum = 45
```

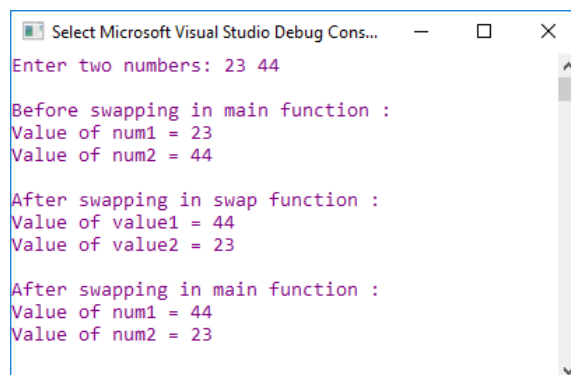
[Estimate Finish Time: 15 minutes]

2. Given the below function prototype:

```
/* function prototype */
void swap(int * value1, int * value2);
```

Write a program to swap two numbers using call by reference.

Output sample:



```
Select Microsoft Visual Studio Debug Cons...
Enter two numbers: 23 44

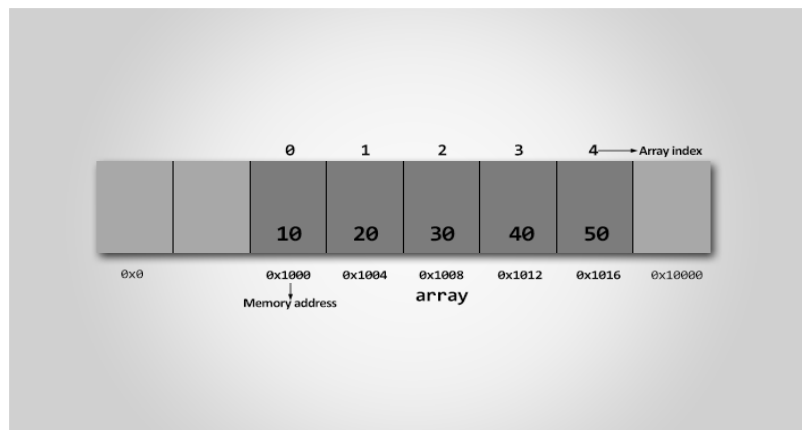
Before swapping in main function :
Value of num1 = 23
Value of num2 = 44

After swapping in swap function :
Value of value1 = 44
Value of value2 = 23

After swapping in main function :
Value of num1 = 44
Value of num2 = 23
```

[Estimate Finish Time: 15 minutes]

Part B: Learn how to input and print array elements using pointers in C++



1. Write a C++ program to input and print array elements using pointer. The number of input should be determined by the user.

Note: It should be a dynamic allocated array.

Output sample:

Microsoft Visual Studio Debug Console

```

Enter the size of array: 4
Now, build an array with 4 columns.
Enter elements in the developed array:
33
4
22
3
The array elements = 33 , 4 , 22 , 3 ,

```

Microsoft Visual Studio Debug Console

```

Enter the size of array: 6
Now, build an array with 6 columns.
Enter elements in the developed array:
33
2
55
6
33
4
The array elements = 33 , 2 , 55 , 6 , 33 , 4 ,

```

[Estimate Finish Time: 15 minutes]

Part C: Homework.

Submit your answer (*in doc / pdf*) to the Microsoft Teams. Your answer should include your code and your program screenshot. Submission due date: **20 October 2019 (11.59pm)**.

1. Complete the following program skeleton. When finished, the program will ask the user for a length (in inches), convert that value to centimeters and display result. You are to write the function convert. (Note: 1 inch = 2.54cm. Do not modify function main.)

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

// Write your function prototype here

void main ()
{
    double measurement;

    cout << "Enter a length in inches, and I will convert\n";
    cout << "it to centimeters: ";
    cin >> measurement;
    convert(&measurement);
    cout << fixed << setprecision(4);
    cout << "Value in centimeters: " << measurement << endl;
}

// Write your function convert here
```

2. Write a program that will ask the user to enter the width and length of a rectangle, and then display the rectangle's area. The program calls the following functions:
 - getLength – this function should ask the user to enter the rectangle's length, and then return that value as a double.
 - getWidth – this function should ask the user to enter the rectangle's width, and then return that value as a double.
 - getArea – this function should accept the rectangle's length and width as arguments, and return the rectangle's area. The area is calculated by multiplying the length by the width.
 - displayData - this function should accept the rectangle's length, width, and area as reference arguments, and display them in an appropriate message on the screen.
3. Write a program that accept an integer argument indicating the total number of subjects and the test scores are stored in a dynamically allocated array. The program calls the following functions:
 - calcAverage – this function should accept an array argument to calculate the average of the test scores.
 - findLowest – this function should find and return the lowest of test scores array passed to it.

Input validation: Do not accept a negative number.

Exercise 4 – Driver’s License Exam:

Write a program that grades the written portion of the driver’s license exam. The exam has 20 multiple choice questions. Here are the correct answers:

1.	B	6.	A	11.	B	16.	C
2.	D	7.	B	12.	C	17.	C
3.	A	8.	A	13.	D	18.	B
4.	A	9.	C	14.	A	19.	D
5.	C	10.	D	15.	D	20.	A

Your program should store the correct answers shown above in an array. It should ask the user to enter the answers for each of the 20 questions, and the answers should be stored in another array. After the student’s answers have been entered, the program should have a function to display a message indicating whether the student passed or failed the exam. (Note: A student must correctly answer 15 of the 20 questions to pass the exam.)

Input validation: Only accept the letters A, B, C, or D as answers.