

GE211 Programming in C++

Lab (10)

Objectives of this lab:

- Learn about Pointers.
- Learn about Dynamic Variables and Dynamic Arrays.

Exercise 1 – Pointers Introductions:

- Compile and run the program and see what it does.
- Add a dynamic variable using: `p1 = new int`; Examine the output of it when:
`cin >> *p1;`
`*p1 = *p1 + 7;`

```
//Lab10_01.cpp. An Introductions to Pointers
#include<iostream>
using namespace std;

int main ()
{
    int    num;
    char   c;
    int    count;
    float  salary;

    // Declaring bunch of pointers that point to nothing (point to null)
    int    *numptr;      // declare a pointer variable to an integer
    char   *cptr;        // declare a pointer variable to a character
    int    *countptr;    // declare a pointer variable to an integer
    float  *salaryptr;   // declare a pointer variable to a float

    // Making them point somewhere
    numptr = &num;  //numptr is pointing to num
    cptr = &c;    //cptr is pointing to c
    countptr = &count;  //countptr is pointing to count
    salaryptr = &salary;  //salaryptr is pointing to salary

    cout << "The address stored in numptr   = " << numptr << endl;
    cout << "The address stored in cptr     = " << (void*) cptr << endl;
    cout << "The address stored in countptr = " << countptr << endl;
    cout << "The address stored in salaryptr= " << salaryptr << endl;
    cout << endl;

    // Assign values to the locations where the pointers point
    *numptr = 2;
    *cptr = 'A';
    *countptr = 100;
    *salaryptr = 3200;

    // Display the content of what pointers variable point to
    cout << "The value pointed by numptr   = " << *numptr << endl;
    cout << "The value pointed by cptr     = " << *cptr << endl;
    cout << "The value pointed by countptr = " << *countptr << endl;
    cout << "The value pointed by salaryptr= " << *salaryptr << endl;
    cout << endl;
```

```

        ++*numptr;
        ++*cptr;
        ++*countptr;
        ++*salaryptr;
        // Display the content of what pointers variable point to
        cout << "The value pointed by numptr   = " << *numptr << endl;
        cout << "The value pointed by cptr     = " << *cptr << endl;
        cout << "The value pointed by countptr = " << *countptr << endl;
        cout << "The value pointed by salaryptr= " << *salaryptr << endl;
        cout << endl;

        // Display the contents of those memory locations
        cout << "The value of num   = " << num << endl;
        cout << "The value of c     = " << c << endl;
        cout << "The value of count = " << count << endl;
        cout << "The value of salary= " << salary << endl;
        cout << endl;

        ++numptr;
        cout << "The address stored in numptr   = " << numptr << endl;
        cout << "The value pointed by numptr   = " << *numptr << endl;
        cout << endl;

    return 0;
}

```

Exercise 2 – Pointers Introductions:

- Compile and run the program and see what it does.
- Test the output when we change `*(listptr++)` to `*listptr++` in line 25.
- Could you explain the difference when using:

```

//Lab10_02.cpp. Array and Pointer
#include <iostream>
using namespace std;

int main(void)
{
    int numlist[5];
    int *listptr;

    // Read 5 integers from the keyboard
    for (int i = 0; i < 5; i++ )
    {
        cout << "Enter value #" << i << ": ";
        cin >> numlist[i];
    }

    // Display the array using its reference:
    for (int i = 0; i < 5; i++ )
        cout << "Value #" << i << ": " << *( numlist + i) << endl;

    // Display the array using the pointer:
    listptr = numlist;

    for (int i = 0; i < 5; i++ )
        cout << "Value # (pointer style) " << i << ": " << *(listptr++) << endl;
}

```

```

    return 0;
}

```

Post-Lab:

Make a modification to Exercise 2, so that it creates two different sizes of dynamic array during the run-time. The size of this array should be entered from the user during the program run-time. Use:

```

    int size;
    cout << " enter the size of the dynamic array " << endl;
    cin >> size;

    int *a = new int[size];
    cout << endl << " enter " << size << " values of the dynamic array " << endl;
    for (int i = 0; i < size; i++ )
    {
        cout << "Enter value #" << i << ": ";
        cin >> a[i];
    }
    // Display the dynamic array:
    cout << endl << " Display the dynamic array: " << endl;
    for (int i = 0; i < size; i++ )
        cout << "Value #" << i << ": " << *( a + i) << endl;

    delete a;
    cout << endl << " enter another size of the dynamic array " << endl;
    cin >> size;
    a = new int[size];
    cout << endl << " enter " << size << " values of the dynamic array " << endl;
    for (int i = 0; i < size; i++ )
    {
        cout << "Enter value #" << i << ": ";
        cin >> a[i];
    }
    // Display the dynamic array:
    cout << endl << " Display the new dynamic array: " << endl;
    for (int i = 0; i < size; i++ )
        cout << "Value #" << i << ": " << *( a + i) << endl;

    return 0;
}

```