**Lab 23 (Intro to Pointers)**

**Lab Task 1.**

Complete the following tasks inside the main function. Consider the type (both pointer and ordinary variable)is integer for all the questions.

1. Declare and initialize an integer variable x with value 10,
2. Declare a pointer p to integer.
3. Point the pointer to x.
4. Display the address of x.
5. Display the pointer p. The output should be the same as step 4.
6. Display x.
7. Display x indirectly (using the indirection operator). The output should be the same as step 6.
8. Declare two pointers named ptr1 and ptr2 on the Same line.
9. Declare a pointer and an integer variable on the same line.
10. Declare a pointer, an integer and an array variable on the line.
11. Declare and initialize two pointers named ptr1 and ptr2 on the Same line.
12. Declare and initialize a pointer and an integer variable on the same line.
13. Declare and initialize a pointer, an integer and an array variable on the line.

**Lab Task 2.**

Repeat Lab tasks 1 from Lab 15, which makes use of passing parameters by reference. However, the parameters should be now passed by reference using the pointer mechanism. For your reference, a parameter is passed by reference to a function as follows:

**Function prototype:** void myFunc(VarType\* ptr);

**Function call:** myFunc(&var); // where ‘var’ has data type = ‘VarType’

You can modify the actual argument ‘var’ by assigning value to ptr using the indirect referencing (derference operator) mechanism, i.e. \*ptr.

**PRACTICE TASKS:**

Repeat Lab tasks 2 from Lab 15, which makes use of passing parameters by reference. However, the parameters should be now passed by reference using the pointer mechanism.

**(b)**

Given the declaration:

int x;

int \*p;

int \*q;

Mark the following statements as valid or invalid. If a statement is invalid, explain why.

a. p = q;

b. \*p = 56;

c. p = x;

d. \*p = \*q;

e. q = &x;

f. \*p = q;

**(c)**

What is the output of the following C++ code?

int x;

int y;

int \*p = &x;

int \*q = &y;

x = 35;

y = 46;

p = q;

\*p = 78;

cout << x << " " << y << endl;

cout << \*p << " " << \*q << endl;