Higher Diploma in Information Technology

Introduction to Programming (C++)

Year 1 Semester 1 – 2023

Worksheet 09

Intended Learning Outcomes:

At the end of the class the students should be able to:

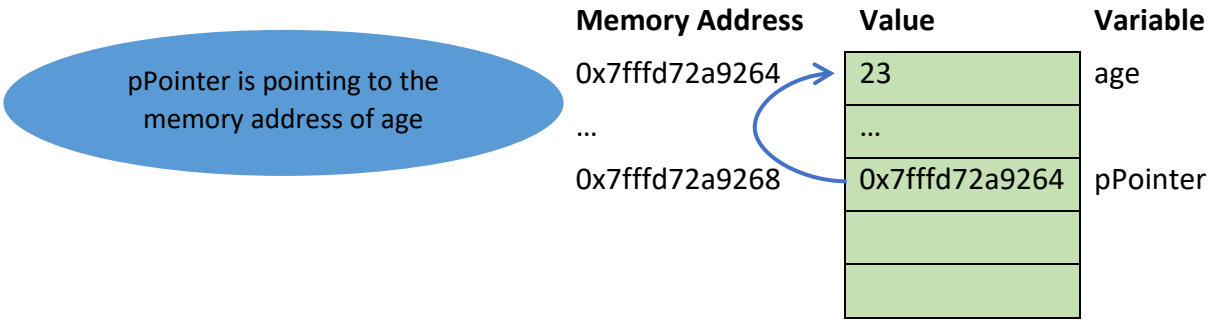
- To learn how to use pointers in C++ programs.
- To use pointers with arrays in C++ programs.
- To learn the command line arguments in C++ programs.

Pointer is a variable that stores a memory address.

Exercise 01: The program below shows how an indirect reference can be made. Compile the below program and analyze the output. Save the program with the name of **Ex01.cpp.**

Sample Output

```
Memory address of pPointer: 0x7fffd72a9268
Memory address of age variable: 0x7fffd72a9264
Integer value of age: 23
pPointer is pointing to 0x7fffd72a9264
Value pointed by pPointer: 23
Now age is 25 through the pointer 0x7fffd72a9264
Value of newAge is 25 Memory address of newAge is 0x7fffd72a9260
```



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

int main()
{
   // Declare a pointer.
   int  *pPointer ;
   int age = 23;

   /* Use address of operator & to assign a memory address of
   age to a pointer */
   pPointer = &age;

   cout<< "Memory address of pPointer: " << &pPointer <<endl;
   cout<< "Memory address of age variable: " << &age <<endl;
   cout<< "Integer value of age: " << age <<endl;

   cout<< "pPointer is pointing to " << pPointer <<endl;
   cout<< "Value pointed by pPointer: " << *pPointer <<endl;

   // modifying actual variable using the pointer
   *pPointer = 25;
   cout<< "Now age is "<<age<<" through the pointer " << pPointer
<<  endl;

   //assign the value pointed by pPointer to newAge variable
   int newAge = *pPointer;
   cout  <<"Value  of  newAge  is  "<<newAge<<"  Memory  address  of
newAge is "<<&newAge<<endl;
   return 0;
}
```

Exercise 02: Write a C++ program to add two numbers using the pointers in C++. Save the program with the name of **Ex02.cpp.**

Exercise 03: Write a C++ program to Increment and Decrement Integer Using Pointer in C++. Save the program with the name of **Ex03.cpp.**

Exercise 04: Write a C++ program to obtain the following output. Save the program with the name of **Ex04.cpp.**

Sample Output

```
Element Value        Memory Address
0        65             0x70fde0
1         5             0x70fde4
2         7             0x70fde8
3         8             0x70fdec
4        45             0x70fdf0
5         7             0x70fdf4
6        25             0x70fdf8
Memory address of marks[]: 0x70fde0
Memory address of ptrMarks: 0x70fdd8
Value pointed by ptrMarks: 65
```

Exercise 05: Write a C++ program find a maximum and minimum value of the array using the pointers in C++. Save the program with the name of **Ex05.cpp.**

```
Array elements :
10 30 50 100 90

Minimum value of array is:10
Maximum value of array is:90
```

Exercise 06: Command Line Arguments in C++

The program will pass the command line arguments to the main () function.

In C / C++, the main () function takes in two additional parameters for these arguments.

```
int main(int argc, char* argv[]);
```

argc	Argument Count. Gives the number of arguments that we pass (includes the program name also)
argv	Argument Vector. This is a char array of strings. These are the argument values itself.

Following program is to accept a text and two integers from command line and then prints total of two integers.

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

int main(int argc, char* argv[])
{
        int no1, no2, sum;
        string text;

        cout << "argc = " << argc << endl;
        if(argc!=4)
                cout<< "Invalid number of args\n";

        else
        {
           text = argv[1];
           no1 = atoi(argv[2]);
           no2 = atoi(argv[3]);
           sum = no1 + no2;
           cout<<"The text is "<<text<<" and Total is "<<sum<<endl;
         }
        return 0;
}
```

atoi() *converts the ASCII value into an integer, since the command line arguments are read as strings.*

Save the program with the name of **Ex06.cpp.**