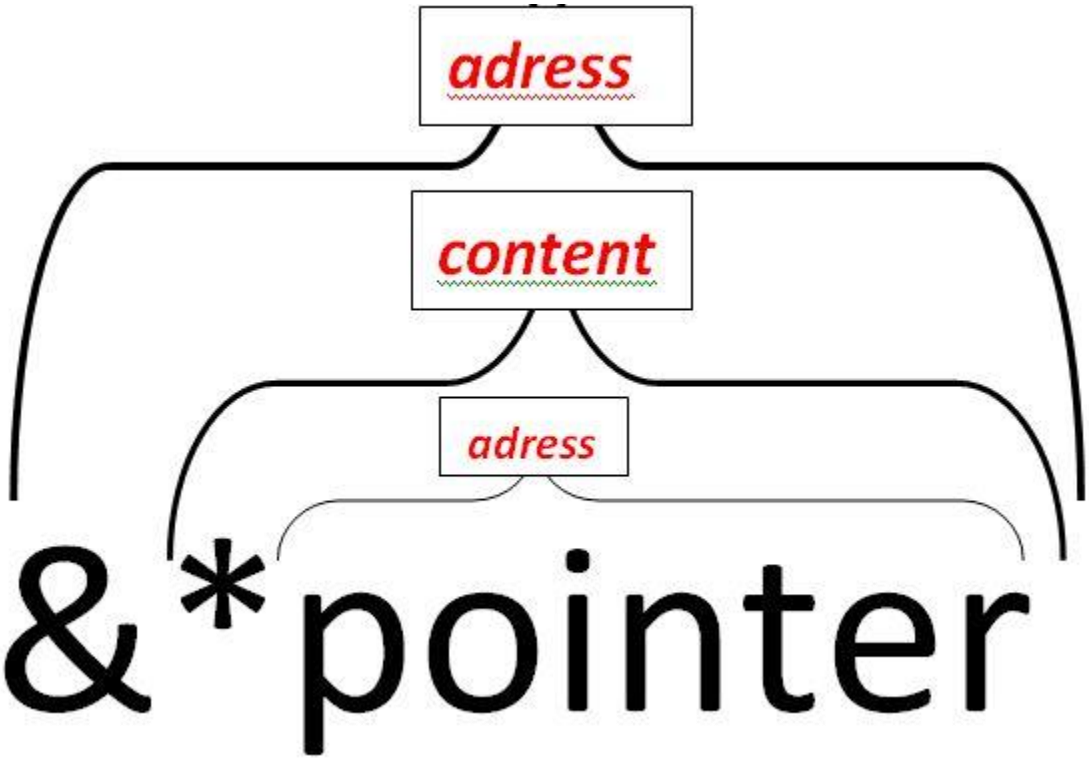
UNIVERSITY OF ENGINEERING AND TECHNOLOGY, TAXILA

FACULTY OF TELECOMMUNICATION AND INFORMATION ENGINEERING

COMPUTER ENGINEERING DEPARTMENT

Data Structure and Algorithms

Pointers in C++



Dated:

Semester:

Autumn 2020

Objectives:-

The objectives of this session is to understand the concept of pointers and how to use pointers for efficient programming in C++.

Introduction:-

To understand the pointers we need to have the knowledge of:

1. Memory Locations
2. Memory Addresses
3. Storage of variables in memory.

Memory Addresses and Variables:-

Computer memory is divided into various locations. Each location is of one byte and each byte has a unique address. When a variable is created into memory three properties are associated with it:

1. Type of variable
2. Name of Variable
3. Memory Address assigned to the variable

e.g.

```
int xyz=6760;
```

where,

int	is the data type
xyz	name of the variable
6760	value of the variable

When a variable is declared a memory location is assigned to it. Suppose the memory address assigned to the above variable xyz is 1011.

To print the contents of the variable xyz we have to write:

```
cout<<xyz<<endl;
```

The memory address where the contents of the specific variable are stored can also be accessed. The address operator (&) is used with the variable name to access its memory address. The address operator (&) is used before the variable name.

For example to print the address of the variable xyz we have to write it like:

```
cout<<&xyz<<endl;
```

The memory address is printed in hexadecimal format.

Pointer Variable:-

The variable that is used to hold the memory address of another variable is called a pointer variable or simply a pointer.

The data type of the variable (whose address a pointer is to hold) and the pointer variable must be the same.

A pointer variable is declared by placing a asterisk (*) after data type or before variable name in data type statement. E.g. if pointer variable “p” is to hold memory address of an integer variable it is declared as:

```
int *p;
```

or to hold address of a float type variable we can declare as:

```
float *rep;
```

Example No 01:-

Write a program to assign two values to two integer type variables a and b. Assign the memory address of variable a and b to pointer variable x and y respectively. Print out the memory addresses of the variables and b through their pointer variables.

```
#include <iostream>

#include <cstdlib>

using namespace std;

int main()

{
    int a,b,*x,*y;
    a=126;
    b=19;
    x=&a;
    y=&b;
    cout<<"Memory Address of Variable a = "<<x<<endl;
    cout<<"Memory Address OF Variable b = "<<y<<endl;
    system("PAUSE");
    return 0;
}
```

We can access the contents of the memory addresses of a pointer variable by placing a asterisk (*) before pointer variable. e.g. to access contents of a and b through pointer variable x and y a asterisk (*) is used before pointer variable like:

```
    cout<<"Memory Address of Variable a = "<<x<<endl;
    cout<<"Memory Address OF Variable b = "<<y<<endl;
```

Example No 02:-

Write a program to assign a value to a variable using its pointer variable. Print out the value using variable name and also print out the memory address of the variable using pointer variable.

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main()
{
    int a,*x;
    x=&a;

    cout<<"Enter the data value ?"<<endl;

    cin>>*x;

    cout<<"Value of Variable is = "<<a<<endl;

    cout<<"Memory Address OF Variable = "<<x<<endl;

    system("PAUSE");

    return 0;

}
```

“Void” type Pointer:-

```
     void *p;
```

The pointer variable “p” can hold the memory address of the variables of any data type.

Pointers and Arrays:-

There is a close relationship between pointers and arrays. In advanced programming arrays are accessed using pointers.

An array consist of consecutive locations. To access array the memory location of the first element of the array is accessed using pointer variable. Pointer is then incremented to access other elements of the array. The pointer is increased in value according to the size of the elements of the array.

```
     int x[5];

     int *p;
```

To store the starting address of the array x the following statement is used.

```
     p=x;
```

& operator is not used only when the array name is used. If the element of the array is used then & operator is used. For example

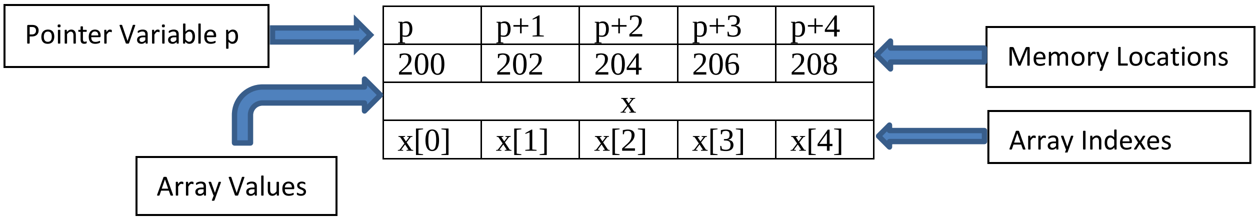
```
     P=&x[0];
```

If one is added to or subtracted from the pointer variable “p” the contents of the pointer variable “p” is incremented or decremented by (1 x size of the object or element) i.e. it is incremented by the size of the object or element to which pointer refer.

For example if location of first element in memory is 200 i.e. the value of the pointer variable “p” is 200 and it refers to an integer variable. When following statement is executed:

```
p=p+1;
```

New value of “p” will be **200 + 1 x 2=202**. All elements of array can be accessed by this technique.



p	p+1	p+2	p+3	p+4
200	202	204	206	208
x				
x[0]	x[1]	x[2]	x[3]	x[4]

Question No 03:-

Write a program to input data into an array and then to print on the computer screen by using pointer notation.

```
#include<iostream>

#include<cstdlib>

using namespace std;

int main()

{

    int arr[5],*pp,i;

    pp=arr;

    cout<<"Enter values into an array" <<endl;

    for(i=0;i<=4;i++)
    cin>>arr[i];
    cout<<"Values from array " <<endl;
    for(i=0;i<=4;i++)
    cout<<*pp++<<endl;
    system("PAUSE");
    return 0;
}
```

Lab Task:-

Write a program to input data into an array and find out the maximum value from array through pointers.

Passing Pointers as Arguments to Functions:-

The pointer variables can also be passed to functions as arguments. When a pointer variable is passed to a function the address of the variable is passed to the function. Thus a variable is passed to a function not by its value but by its reference.

Example 4:-

Write a program to pass two parameters to the function to add a constant value of 100 to the passed values using pointers.

```
#include<iostream>
#include<cstdlib>
using namespace std;
int main()
{
    void temp(int*,int*);
    int a,b;
```

```
        a=10;
        b=20;
        temp(&a,&b);
        cout<<"Value of a = "<<a<<endl;
        cout<<"Value of b = "<<b<<endl;
        cout<<"Okay"<<endl;
        system("PAUSE");
        return 0;
}
void temp(int *x, int *y)
{
         *x=*x+100;
         *y=*y+100;
}
```

In the above program the function "temp" has two parameters which are pointers and are of int type. When the function "temp" is called the addresses of the variables "a" and "b" are passed to the function.

In the function a value 100 is added to both variables "a" and "b" through their pointers. That is the previous values of variables "a" and "b" are increased by 100. When the control returns to the program the value of variable a is 110 and that of variable b is 120.

Lab Task:-

1. Write a program to swap two values by passing pointers as arguments to the function.
2. Write a program to convert Fahrenheit temperature to Celsius degree by passing pointers as arguments to the function.

3. Write a program to convert kilogram into grams by passing pointers as arguments to the function.

Pointers as Strings:-

A string is a sequence of characters. A string type variable is declared in the same manner as an array type variable is declared. This is because the string is an array of character type variables.

Example 5:-

Write a C++ code to print a string character by character.

```
#include<iostream>

#include<cstdlib>

using namespace std;

int main()

{

    char st[]="Pakistan";

    void ppp(char*);

    ppp(st);

    cout<<endl;

    system("PAUSE");

    return 0;

}

void ppp(char *sss)

{

     while(*sss!='\0')

     {
```

```
                      cout<<*sss<<endl;
                      *sss++;
     }
}
```

Example No 06:-

Write a program to find length of string by using pointers.

```
 #include<iostream>
#include<cstdlib>
//#include<string.h>
using namespace std;
int main()
{
    char st[25];
    int x;
    int len(char*);
    cout<<"Enter the String"<<endl;
    gets(st);
    x=len(st);
    cout<<"Length of string is "<<x<<endl;
    cout<<endl;
    system("PAUSE");
    return 0;
}
```

```
int len(char *sss)
{
    int c=0;
     while(*sss!='\0')
     {
                     c++;
                     *sss++;
     }
     return c;
}
```

Home Assignment:-

Write a program to copy one string to another string by using pointers.

Write a program to combine two strings using pointers.