Experiment No 6

Aim: Implementing pointers in CPP

Theory:

Pointers in C++ are a unique type of variable that stores the address of another variable. That is, whenever we declare a variable in CPP programs, it is stored in the memory at a specific address. A pointer to that variable points to the address of the respective variable. In other words, the value stored inside a pointer is the address of the variable it points to.

Just like variables, functions, etc., pointers in C++ also have data types. For example, a pointer of data type is integer that can store the address of the variable of int data type. Similarly, a pointer of data type float can store the address of the variable of float data type.

A pointer is a variable that stores the memory address of an object. Pointers are used extensively in both C and C++ for three main purposes:

- to allocate new objects on the heap,
- to pass functions to other functions
- to iterate over elements in arrays or other data structures.

Pointer Declaration:

data_type *pointer_name;

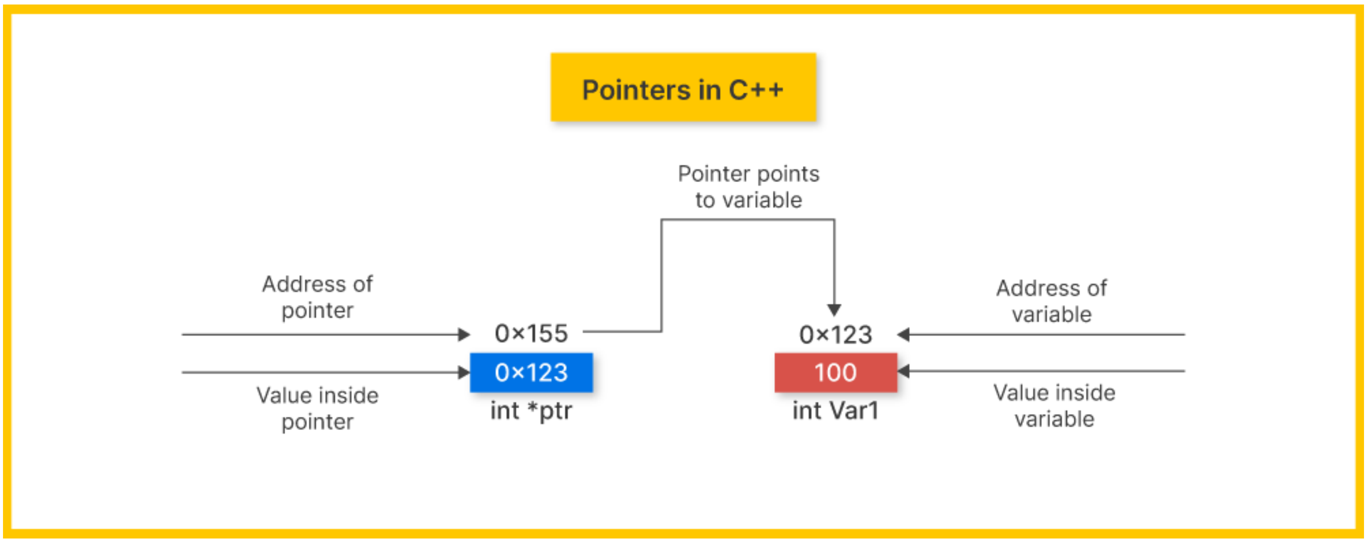
Here,

- Data_type refers to the type of the value/ data the pointer is pointing to.
- The unary operator (*), also known as the dereference operator, indicates that the variable is a pointer. It is essential when declaring a pointer.
- The term pointer_name is the placeholder for the name of the pointer you are creating.

Examples:

Syntax Examples for Pointer Declaration In C++

```
int *ptr; // Pointer of data type int
float *ptr; // Pointer of data type float
char *ptr; // Pointer of data type char
double *ptr; // Pointer of data type double
```



Different Types Of Pointers In C++

There are multiple different types of pointers in cpp classified on the basis of their nature, behaviours, use, values they point to, etc. The most commonly used type of pointers in C++ are-

1. Null Pointers
2. Dangling Pointers
3. Void Pointers
4. Wild Pointer
5. Invalid Pointers

References & Pointers In C++

The purpose of references and pointers in C++ is to provide access to another variable. Both of these are created using the ampersand symbol (&), also called the address-of operator. It hence becomes difficult to differentiate between the two. Here is a brief description of both to provide a clear picture.

Pointers: Pointers in C++ are a special type of variable that stores the memory address of another variable. For example:

int var=10;

int *ptr=&var

References: On the other hand, references are used to create aliases names for a variable declared previously with another name. This allows us to refer to a single variable with multiple names without creating copies of the variable.

int var=10;

int &ref = var;

Irrespective of this, pointers and references are closely related since both allow us to access the data stored in another variable/ location indirectly. Think of it like this- pointers also act as references to a variable by allowing access to its location. Both references and pointers in C++ are widely used to pass arguments to functions . The relation between references and pointers can also be understood by looking at the ways of passing arguments to a function. These include:

- Call by value
- Call-By-Reference with Pointer Argument
- Call-By-Reference with Reference Argument

Practical Related Questions:

1. Explain concept of Pointers to Pointers.
2. Write advantages of pointers.
3. What do the asterisk (*) and ampersand (&) symbols indicate in the pointer in C++?
4. Can a pointer point to two addresses? Justify your answer with proper example.

Programs:-

1. Accept a number from user and print its value and address using pointer.
2. Write a program to exchange the value of two variable using pointer.

Conclusion :-

Hence, we learnt to implementation of Pointers in CPP.

Practical No 6

1. **Accept a number from user and print its value and address using pointer.**

```
#include<iostream.h>
#include<conio.h>
void main()
{
        Int n,*p;
        p=&n;
        clrcsr();
        cout<<"Enter number :";
        cin>>n;
        cout<<"\n number is: "<<*p;
        X=*p;
        cout<<"\n number is :"<<x;
          cout<<"\n Its address is :"<<p;
        getch();
}
```

Q2. Write a rogram to exchange the value of two variable using pointer.

```
#include<iostream.h>
#include<conio.h>
void main()
{
        int a,b,c,*p1,*p2,*p3;
        p1=&a;
        p2=&b;
        p3=&c;
        cout<<"enter two number: ";
        cin>>*p1>>*p2;
        cout<<"\n Before swapping a:  "<<*p1<<"\n b:"<<*p2;
        *p3=*p1;
        *p1=*p2;
        *p2=*p3;
        cout<<"\n After swapping a: "<<*p1<<"\n b:"<<*p2;
        getch();
}
```