





int \*ip; // pointer to an integer

double \*dp; // pointer to a double

float \*fp; // pointer to a float

char \*ch // pointer to character

**Reference operator (&)** as discussed above gives the **address** of a variable.  
To get the **value**stored in the memory address, we use the**dereference operator (\*)**.  
For example: If a number variable is stored in the memory address **0x123**, and it contains a value**5**.  
The**reference (&)** operator gives the value **0x123**, while the **dereference (\*)** operator gives the value **5**.

#include <iostream>

using namespace std;

int main () {

int var = 20; // Variable declaration.

int \*ip; // pointer variable

ip = &var; // store address of var in pointer

// print value inside the variable

cout << "Value of var variable: "<<var<<endl;

// print address of variable var

cout << "Address of var variable: "<<&var<<endl<<endl;

// print value inside the pointer ip

cout << "Value inside pointer ip: "<<ip<<endl;

// print value of variable using pointer ip

cout << "Variable value using Pointer: "<<\*ip;

return 0;

}

Pointers can be used in different ways to perform different types of task. Some uses, variations and topics of pointer are as follows:

* Pointer to Arrays
* Pointer to Pointer
* Pointer to Functions
* Pointer with structures
* Dynamic Memory Allocation
* Null Pointer