Dereferencing Operator

Learn how to get the value pointed out by the pointer.

We'll cover the following

- Indirection/dereferencing operator
  - Example program
  - Explanation

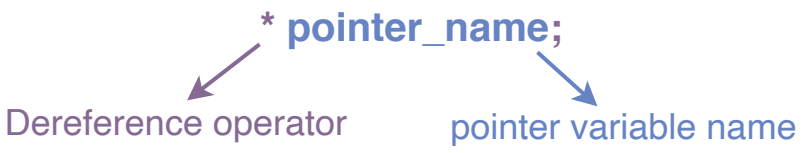
Indirection/dereferencing operator#

Consider the example given in the previous lesson. `John` 's storage house is pointing to `Alice` 's storage house, so `John` is a **pointer** here. What if `John` wanted to know what value is stored in `Alice` 's house?

For this, we will use the dereference operator `*` before the pointer name to access the value of the variable to which the pointer is pointing.

---

*The **dereference operator** `*` is a unary operator. It gives the value of the variable to which the pointer is pointing. This process is known as dereferencing a pointer.*



Example program

Press the **RUN** button and see the output!

```
#include <iostream>

using namespace std;

int main() {
  // Declares a variable Alice
  int Alice = 5;
  // Declares a pointer variable John that can point to int value
  int *John = nullptr;
  // Stores the address of Alice in John
  John = &Alice;
  // Prints value of Alice
  cout << "Value of Alice = " << Alice << endl;
  // Prints value (address of Alice) of John
  cout << "Value of John = " << John << endl;
  // Prints value of Alice
  cout << "Value of Alice = " << *John << endl;

  return 0;
}
```

```
Output                                                    1.21s

 Value of Alice = 5
 Value of John = 0x7ffd9bd714ec
 Value of Alice = 5
```

Explanation#

Line No. 17: `John` accesses the value stored in `Alice` and prints it to the console. Using asterisk `*` is like going to `Alice` 's house and seeing what she has stored in her house.

❌ Trying to dereference an uninitialized or null pointer generates an error.

Quiz

Q Consider the code given below:

```
char *characterPtr , character = 'a';
characterPtr = &character;
```

What is the value of `*characterPtr` ?

A) Address of character

B) Address of characterPtr

Your Answer

C) a

Explanation

`characterPtr` is pointing to the `character` . By using a dereference

operator * with

`chracterPtr`, we will get the value stored in `character`.

D) Error

Submit Answer

Reset Quiz

Interesting so far? Let's explore how the pointers can be passed to functions.

Back

What is a Pointer?

Next

References

Mark as Completed

Report an Issue