Dereferencing Operator

In this lesson, you will learn how to get the value pointed out by the pointer.

We'll cover the following

Indirection/dereferencing operator

Consider the example given in the previous lesson. Here, `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!

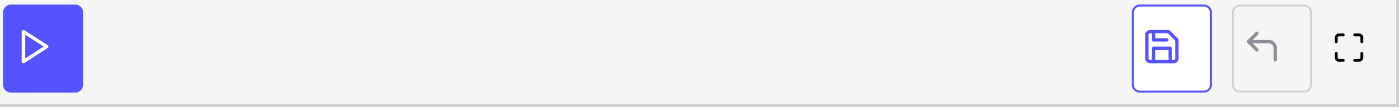
```
#include <iostream>

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;
  // 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;
}
```



Explanation

Line No. 17: Here, `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

Consider the code given below:

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

What is the value of `*characterPtr`?

Retake Quiz

Interesting so far? Let’s dive right in and explore how we can pass the pointers to functions.