Dereferencing a pointer

We learnt how to store an address inside a pointer variable. We can also use the * operator to get the variable stored in a particular address. This process is called **pointer dereferencing**.

For example:

```
int main() {
    int x = 40;     // Declare and initialize an integer variable
    int *p = &x;  // Declare and initialize a pointer to the address
of 'x'

    cout << p << endl;  // Print the address stored in the pointer
    cout << *p << endl; // Print the value stored at the address in
the pointer

    return 0;
}
```

Note that the * symbol can be confusing here, as it does two different things in our code:

- When used in declaration (int *p), it creates a pointer variable.
- When not used in declaration, it act as a dereference operator.

Runt the code in the IDE and check the output.

- The first **`cout`** prints the address that is stored in the pointer **`p`**. This is the memory address where **x** is located.
- Next line uses the **`*`** operator to dereference the pointer **`p`**, which means it accesses the value stored at the address that ptr points to. It prints the value **`40`**, which is the value of **x**.

Click Submit and observe the code and output