Using assert

An assertion is a statement about a condition which must be true when encountered. If the condition is **not true**, then `assert()`, (declared in `<cassert>`), causes the program to immediately fail, printing an error message.

Programmers use assertions to **reason** about **logical correctness**. Assertions can be used to check **preconditions** (what must be true before the program runs correctly), and **postconditions** (what must be true after a calculation completes).

Here is an (admittedly silly) example using `assert()`.

```
cout << "Making sure that 2 + 2 is 5?" << endl;
assert(2 + 2 == 5); // false
```

The programmer assumed (**wrongly**) that the expression `2 + 2` should produce `5`. The assertion causes the program to stop and **print an error message**, so the programmer can fix the mistake. The message will depend on the toolchain. Here is **g++** on Unix.

```
Making sure that 2 + 2 is 5?
a.out: main.cpp:10: int main():
   Assertion `2 + 2 == 5' failed.
Aborted (core dumped)
```

The message includes the executable name (`a.out`), the source file (`main.cpp`), the line number (`10`), the function name and the assertion which failed, so you can immediately open your editor and fix the code.

This course content is offered under a CC Attribution Non-Commercial license. Content in this course can be considered under this license unless otherwise noted.