MCP Summer Workshop

Beginner's C++ Tutorial

Follow along with us!

Table of contents

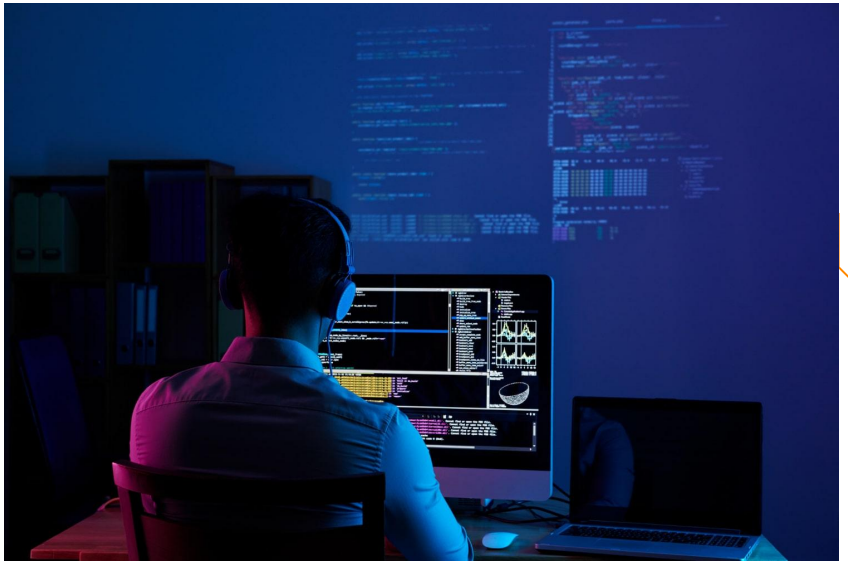
01

Why do we create code?

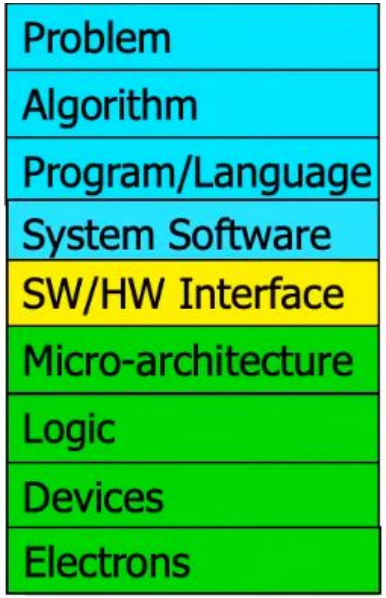
Coming back to this soon...

What are some things that code can accomplish?

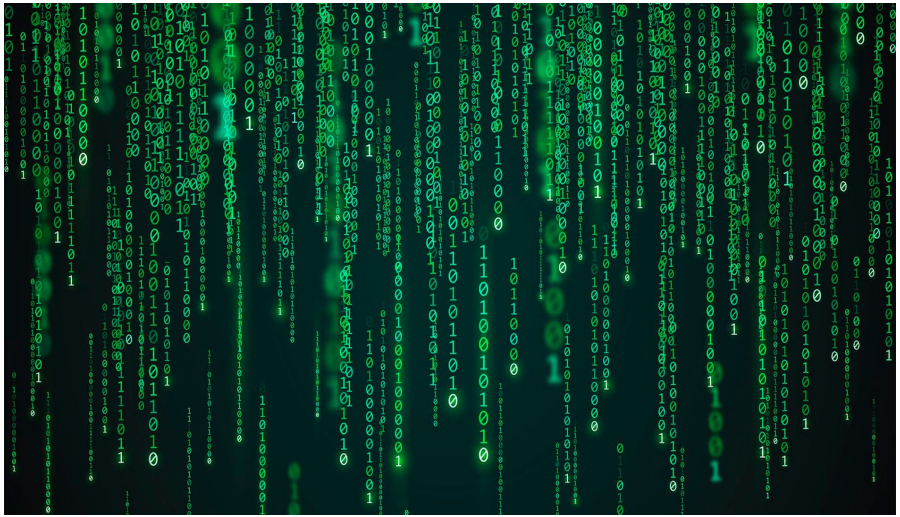
- i.e... moving robots?
- create websites?
- make phone games
- any other examples?



So what is a program?



(Ignore the colours)



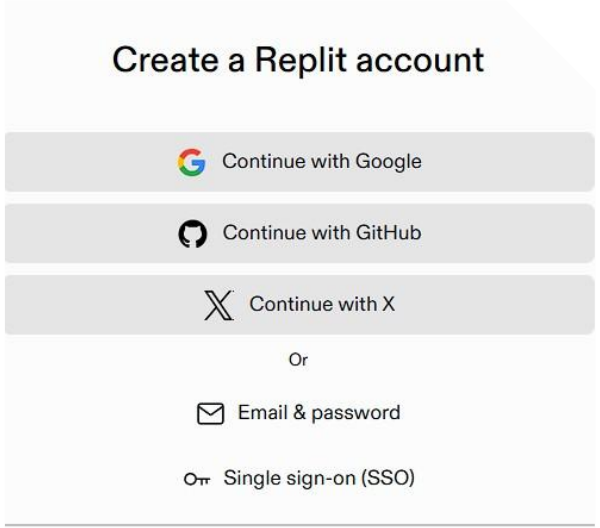
What is the purpose of programming?

Back to the main point...

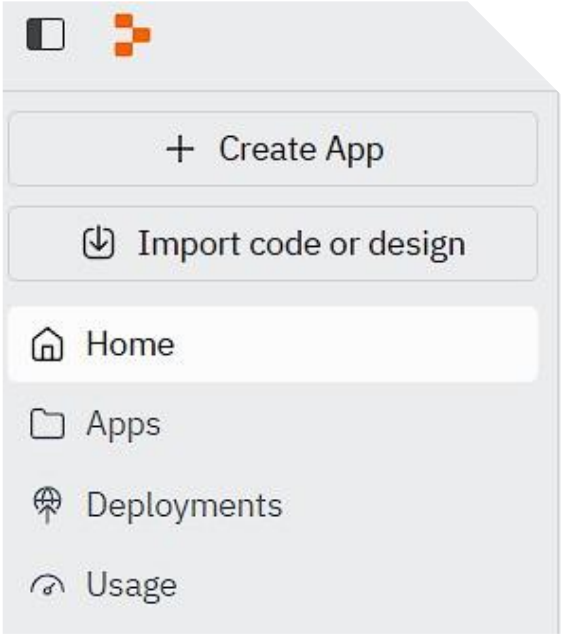
02
Setting Up Your Environment

Very complicated.

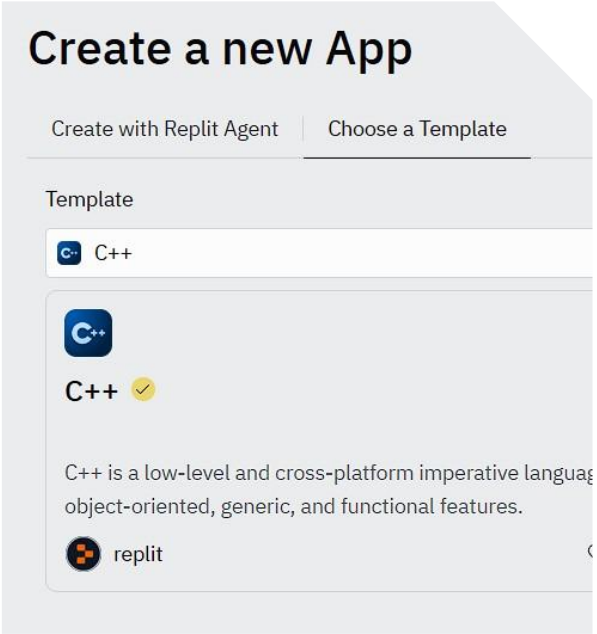
Online IDE: Replit



Sign Up for Replit



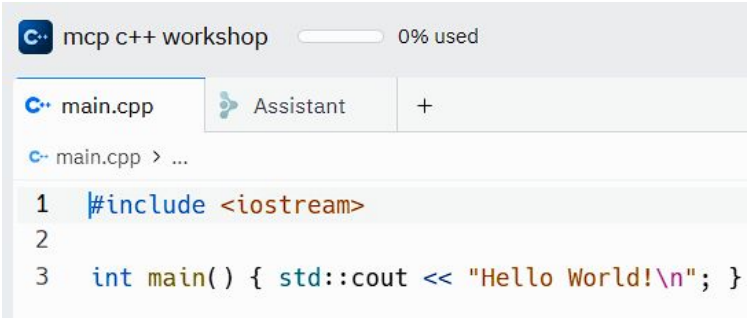
Click Create App



Select the C++ Template

IDEs

- Code Editor, similar to a text editor
- Compiler or Interpreter
- Debugger
- Build Tools
- File/Project Management




```
DLinkedList::DLinkedList() {  // constr
  header = new DNode;         // create
  trailer = new DNode;
  header->next = trailer;  // have them
  trailer->prev = header;
  header->prev = nullptr;
  trailer->next = nullptr;
}
```

Replit offers a cloud based solution
Convenient for small scale projects

OnlineGDB is also an option

We Recommend VScode for larger projects

C++ Under the Hood

1. Source File

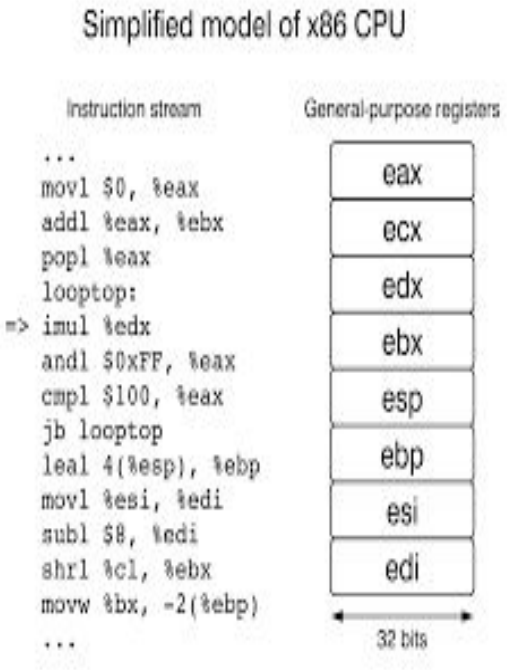
When you write C++ code, you're giving the computer instructions in a language it doesn't fully understand yet. It needs to translate those instructions into something it can actually run — and that's where compiling comes in.

2. Compiler

The compiler takes your .cpp file and translates it into machine code (binary — 0s and 1s).

3. Executable File

To finally execute your program, you must run the executable file which your computer can use.



03

Simple C++

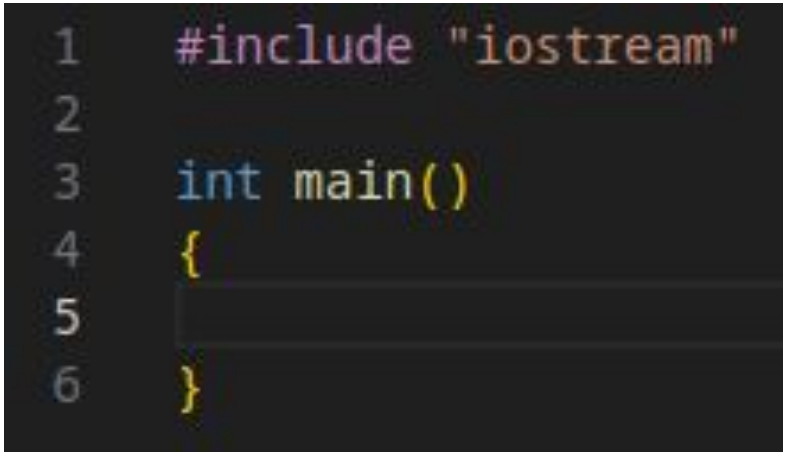
Introducing the basic instructions

Necessities of a C++ Program

Every program has two necessary parts.

- IO -> Input + Output
- int main()

(Basically every program)



int main()

Think of it as a starting line: it tells the computer where to start following your instructions.

Input and Output

Think about it: why does every program need to have some form of output?

Meanwhile, is input necessary for all programs?

```
1   #include "iostream"
2   #include "stdio.h"
3   #include "ncurses.h"
4
```

The First Instruction

Most common 'first program' that people learn; still, there's a lot going on.

Where do the words, "Hello World", actually show up?

```cpp
#include "iostream"

using namespace std;

int main()
{
    cout << "Hello World" << endl;
}
```

```cpp
cout << "Hello, World!" << endl;
```

Special Output Newline

When you want to separate outputs by a new line, there are two separate methods. Keep this in mind!

```
5   int main()
6   {
7       cout << "hi";
8       cout << endl;
9       cout << "bye";
10      cout << "\n";
11  }
```

The Console

A magic mirror.

Write on it, and you shall receive your response…

Fun Fact:

Early “Personal Computers” were much simpler than what we have today. Computer Monitors were limited to a simple ANSI Text display, along with a keyboard for input.

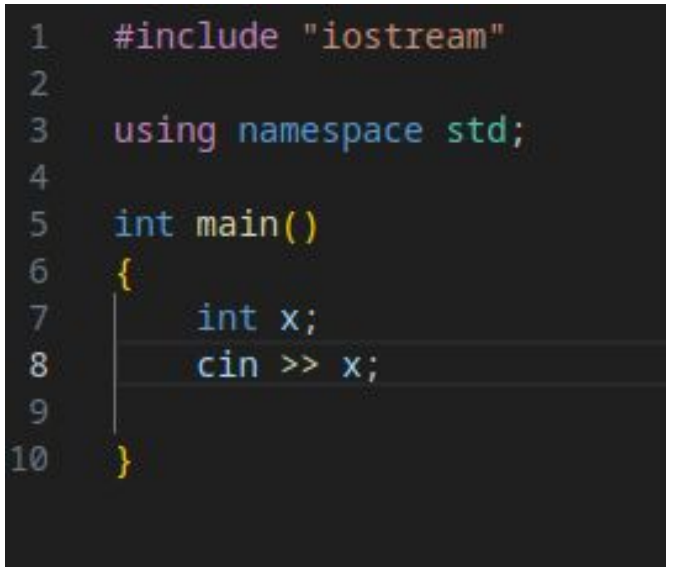
This changed in the 1980’s, where GUI’s (Graphical User Interfaces) were released, developed into the modernized displays of today.

The Complimentary to 'cout'

Now what happens... Nothing!

We have to tell the program what to do with our input first.

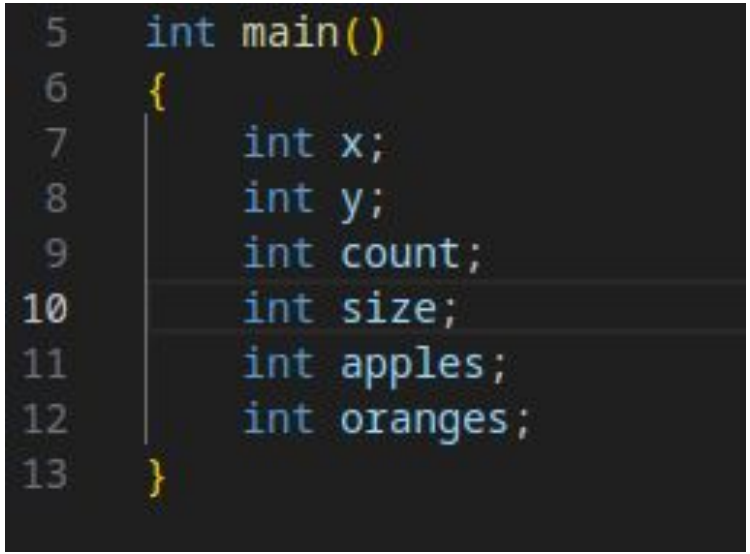
What does 'cin' stand for? Does anybody know?



Integers

But wait, what does "int x" mean?

It's the most fundamental method of storing a piece of information... in this case, an integer value.



Recording an Integer

Nothing fancy.

```cpp
#include "iostream"

using namespace std;

int main()
{
    int x;
    cin >> x;
    cout << x << "\n";
}
```

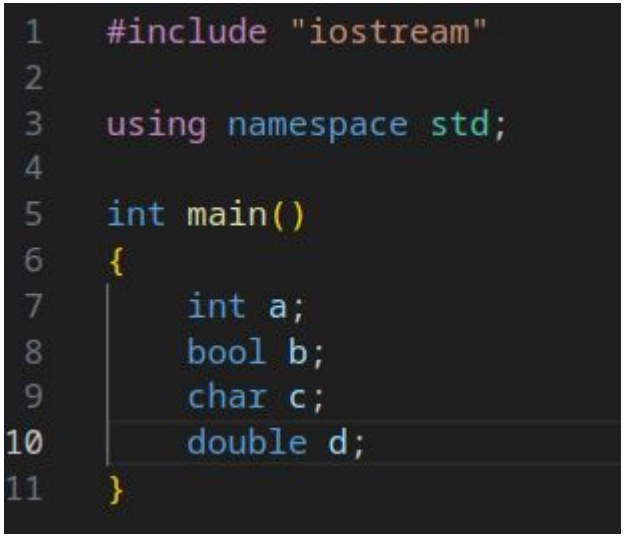
What About.. Characters?

Letters of the alphabet.

Four Fundamental Types

int -> stores an integer value
bool -> stores a boolean value
char -> stores a character value
float/double -> stores a real value

We won't be focusing on reals today! Don't worry about them for now.

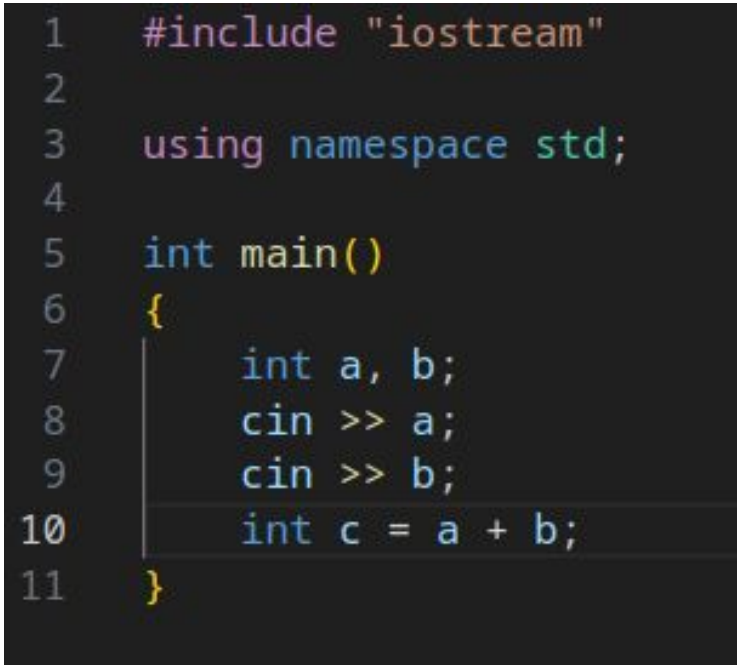


```
#include "iostream"

using namespace std;

int main()
{
    int a;
    bool b;
    char c;
    double d;
}
```

What to do with an integer

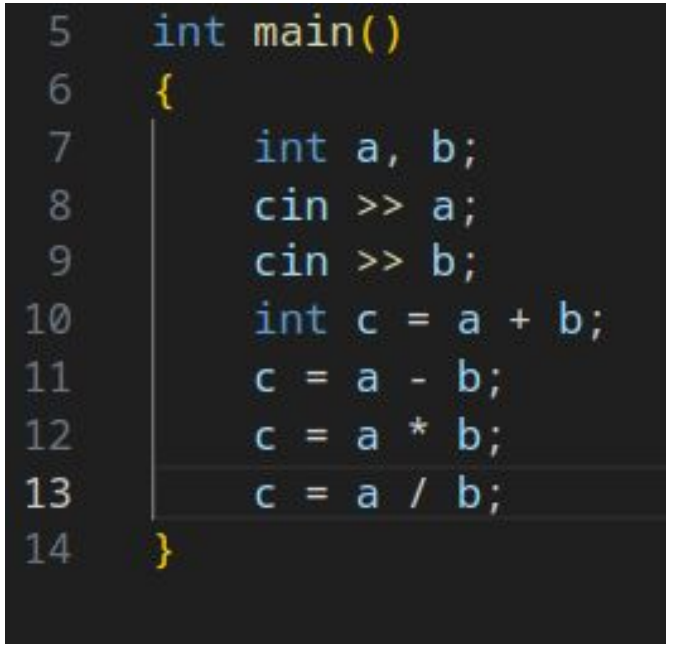
Middle school algebra. And the assignment operator.

The equals sign sets the value of ‘c’ to “a + b”.



Other Basic Operations

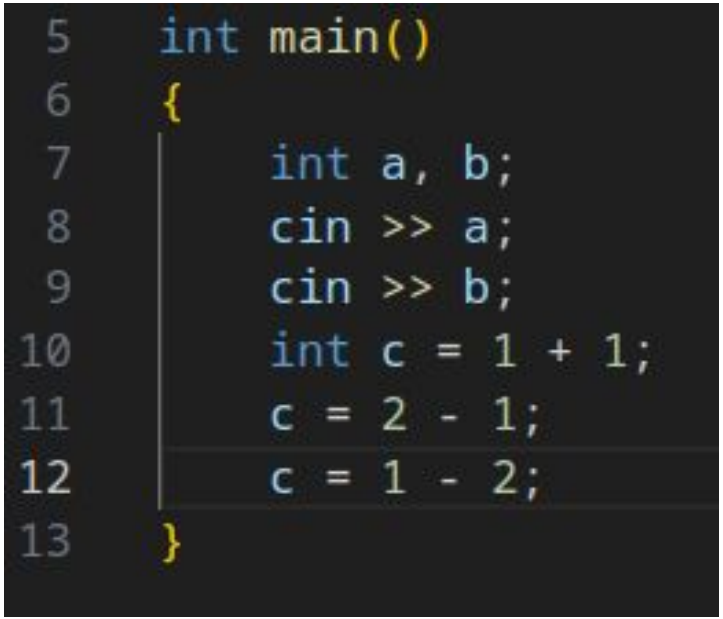
Probably you've seen these symbols before.



```
5   int main()
6   {
7       int a, b;
8       cin >> a;
9       cin >> b;
10      int c = a + b;
11      c = a - b;
12      c = a * b;
13      c = a / b;
14  }
```

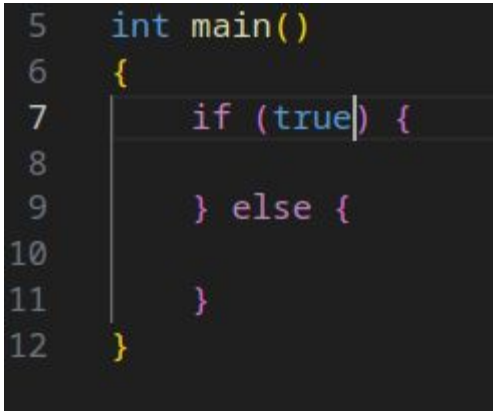
Actual Math

You can use numbers too! Calculations aren't limited to your previous variables.



The Most Important Concept Today

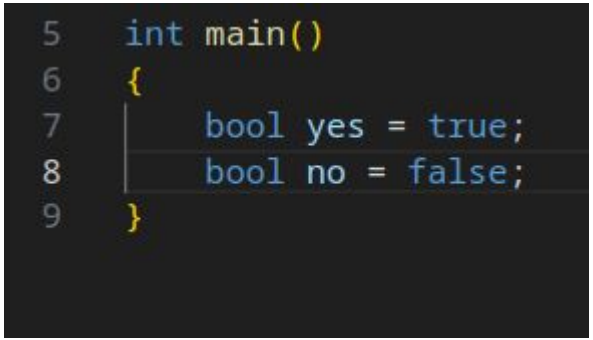
Making decisions.



What's a Boolean?

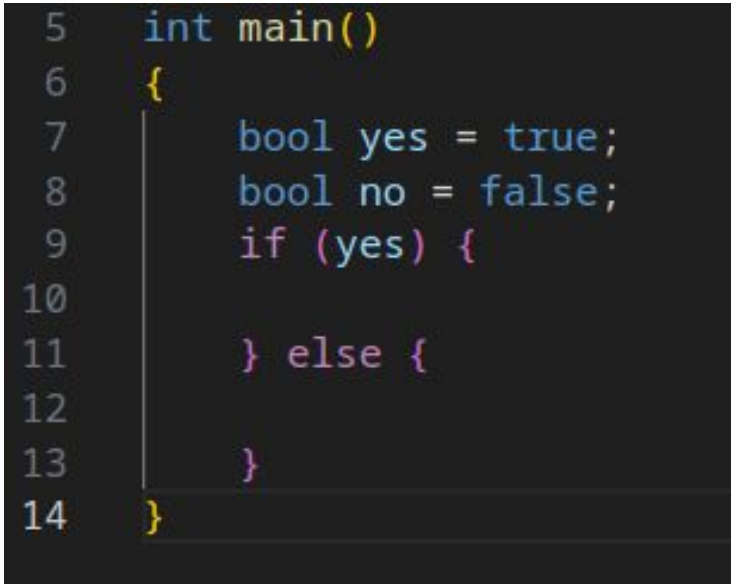
Named after George Boole who introduced the field of boolean algebra.

A bool can only have two values: true or false. There is no other state.



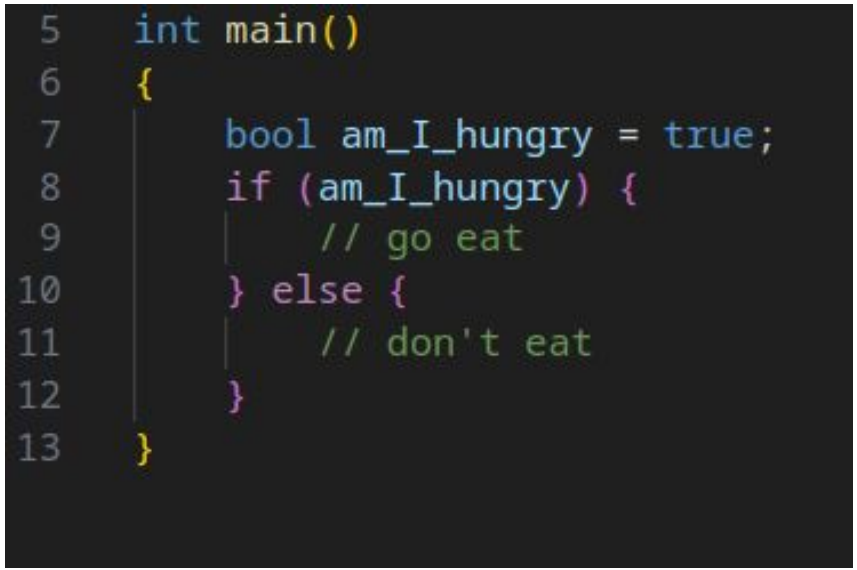
Branches

A program is essentially a plan. We want to plan ahead for any possible changes in this plan to make our program fail-proof.



What Decisions Can We Make

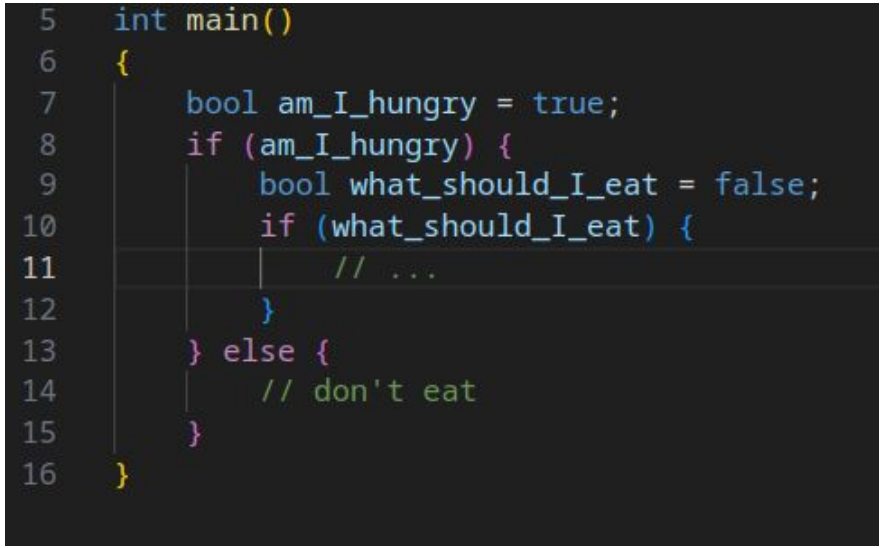
Something I think about many times throughout the day.



Not Everything is Yes or No

A burger? A pie? Some french fries?

True or false doesn't always cut it; we need to think of different ways to ask our question.



```
5   int main()
6   {
7       bool am_I_hungry = true;
8       if (am_I_hungry) {
9           bool what_should_I_eat = false;
10          if (what_should_I_eat) {
11              // ...
12          }
13      } else {
14          // don't eat
15      }
16  }
```

Making Multiple Decisions

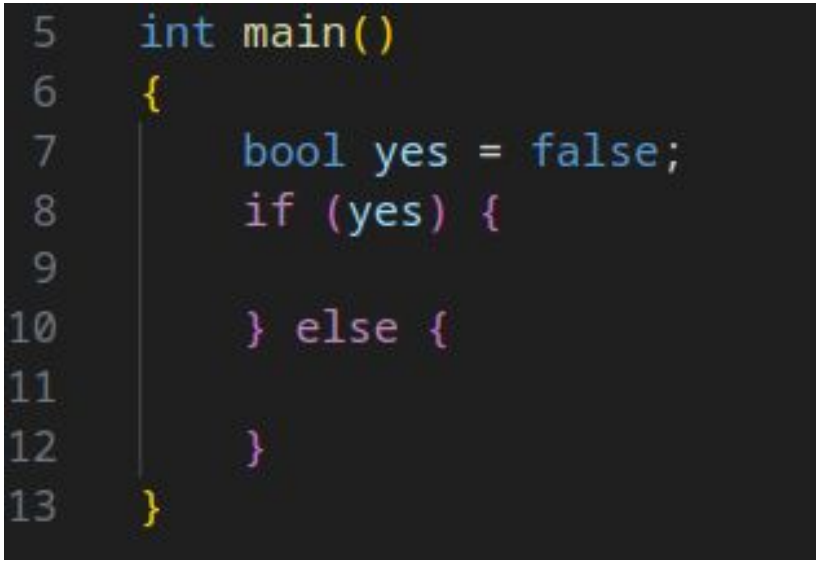
‘if’ blocks can be chained together.

```
5   int main()
6   {
7       bool am_I_hungry = true;
8       if (am_I_hungry) {
9           bool I_want_burgers = false;
10          if (I_want_burgers) {
11              // get some burgers
12          }
13          bool I_want_a_pie = false;
14          if (I_want_a_pie) {
15              // get a pie
16          }
17          bool I_want_a_french_fry = true;
18          if (I_want_a_french_fry) {
19              // get a french fry
20          }
21      }
22  }
```

Preparing For the Worst

What if our initial condition isn't true? What happens then?

Our trusty 'else' block!



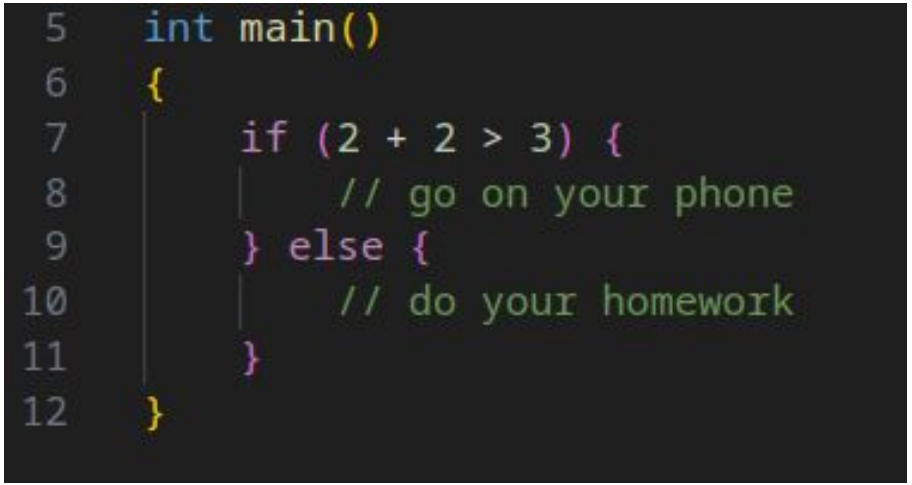
What Decisions Can We Make 2

Know that if you have a conditional, you'll have to account for both true and false options.

```
5   int main()
6   {
7       bool friend_is_available = false;
8       if (friend_is_available) {
9           // play some basketball
10          // play some video games
11          // idk
12      } else {
13          // find something else to do
14      }
15  }
```

Mathematic Conditionals

Sometimes you can't read the user's mind! You'll then have to figure out yourself what to do next.



Simple AI

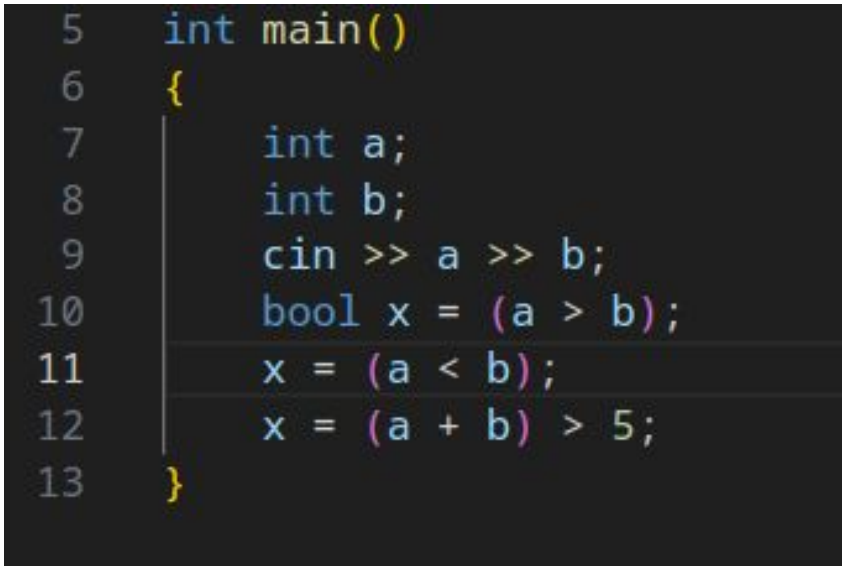
For when you're really tired and don't want to think.

Lines 11 and 12 are separated for readability :)

```
5   int main()
6   {
7       int how_much_money_I_have;
8       int how_much_money_it_costs;
9       cin >> how_much_money_I_have;
10      cin >> how_much_money_it_costs;
11      bool can_I_buy_it = (how_much_money_I_have >=
12          how_much_money_it_costs);
13  }
```

Other Forms of Conditionals

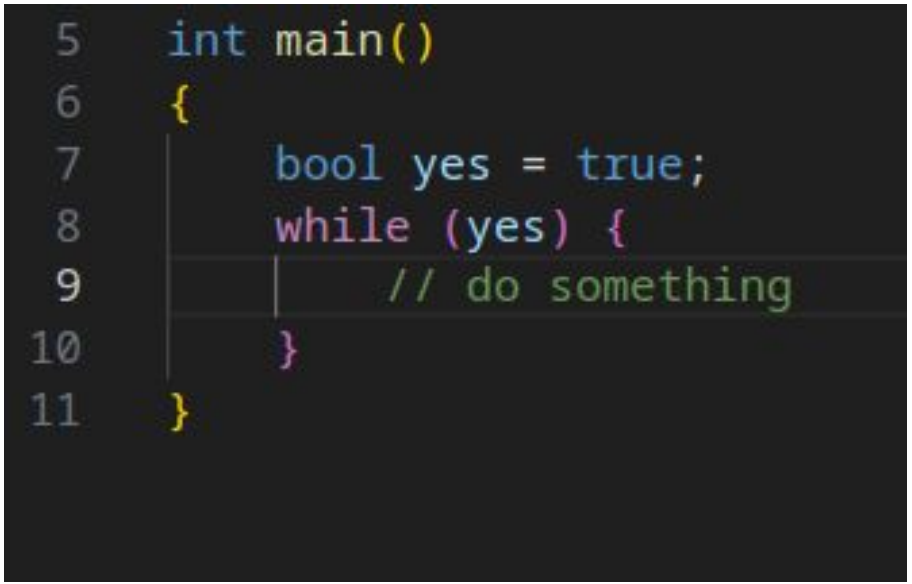
Less than, more than, less than or equal to, more than or equal to. There are many possible conditionals you can put in an if statement.



Loops!

There are two types of loops; we'll cover 'while' loops today.

The other type takes longer to type.



Cheering Up One's Self

Let's analyse this program together!

A variable on the left hand side can refer to itself on the right hand side.

```
5   int main()
6   {
7       int count;
8       cin >> count;
9       while (count > 0) {
10          cout << "you are special.";
11          cout << "\n";
12          count = count - 1;
13      }
14  }
```

Will this Loop Ever End?

Be careful writing your loop conditions!

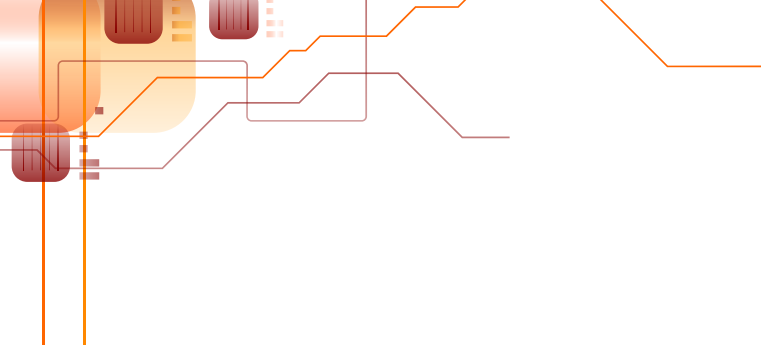
```
5   int main()
6   {
7       int i;
8       cin >> i;
9       while (i > 0) {
10          cout << "our variable is now: ";
11          cout << i << endl;
12          i = i + 1;
13      }
14  }
```

What Have We Learned?

In essence, I've gone through the three fundamental parts of a computer program.

1. Store and retrieve information
2. Gather and create new information
3. Make decisions

```
5   int main()
6   {
7       int steps;
8       int goal;
9       cin >> steps;
10      cin >> goal;
11      if (steps > goal) {
12          cout << "Great Job!";
13      } else {
14          cout << "Keep Going!";
15      }
16  }
```



That's All of the Basics

All you needed to know.. for now...

Next up: our challenge problems!

04

Some Practical Exercises

Make sure you try them out yourself!

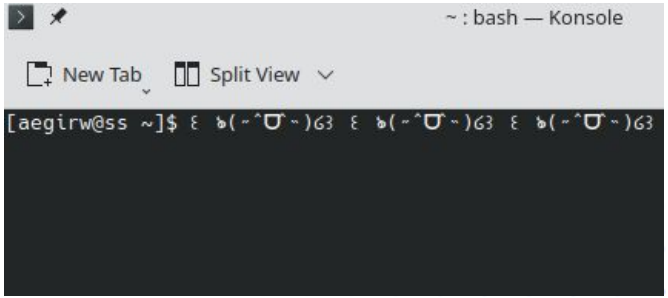
Problem Title

Problem Description.

A paragraph explaining the problem will be shown here. Read through it carefully!

Expected Input/Output

What the program's input/output should look like when your code is correct.



Adding Two Numbers

Your task is simple: receive two integers from the user and add them.

Input: two integers, separated by a space.

Output: the sum of the two integers

Input:

Output:

Solution 1

Adding N Numbers

The user will give you a number. Then, they will give you a list of that size of integers. Your task is to add them all and return the sum.

Input: The first line will contain N, where N is the size of the list of numbers. The second line will contain that list of numbers.

Output: The sum of these numbers

Input

```
3
1 2 3
```

Output

```
6
```

Solution 2

Min-Max Sum

Given three numbers, find the minimum and maximum sum after choosing two of three numbers.
For example, given the three numbers [2, 4, 6], the maximum sum would be 10, and the minimum sum would be 6.

Input: three positive integers

Output: the minimum sum, then the maximum sum.

Input:

```
1 10 3
```

Output:

```
4 13
```

Solution 3

Maximizing Ice Cream

It's a hot summer day out, and you currently have *'A'* ice cream cones. A suspicious man comes up to you and offers you two choices; the first is to receive *'B'* extra ice cream cones, and the second is to multiply your current number of ice cream cones by *'C'*. Which choice maximizes your ice cream cones?

Input: three integers, A, B, and C

Output: B or C depending on the better choice

Input:

```
5 10 2
```

Output:

```
B
```

Solution 4

Maximizing Ice Cream 2

Now, you'll have to solve the problem more than once. The input will start with a number 'N', and you'll receive the three previous integers 'N' number of times.

Input: the number N. Then, on 'N' lines, you'll receive three integers, A, B, and C.

Output: B or C depending on the better choice, for each of the 'N' inputs.

Input:

```
3
5 10 2
5 1 2
3 4 2
```

Output:

```
B
C
B
```

Solution 5

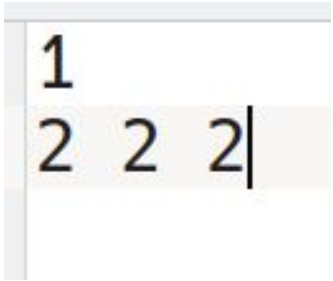
Maximizing Ice Cream 3

What if both choices give the same number of ice creams? We want to deal with the special case!

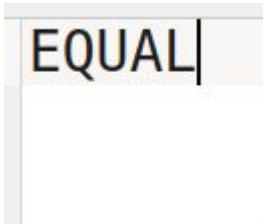
Input: the number N. Then, on 'N' lines, you'll receive three integers, A, B, and C.

Output: B or C depending on the better choice for each of the 'N' inputs. In the case that both choices give equal benefits, output the word "EQUAL".

Input:



Output:



Solution 6

@maccpteam on Instagram :)