Input and Output Functions

Apart from 'cin' and 'cout', there are many other functions we can use to perform input/output operations.

WE'LL COVER THE FOLLOWING ^

- Input
 - Formatted input
 - Unformatted input
- Output
- Further information

Input

In C++ we can read from the input stream in two different ways: Formatted with the extractor >> and unformatted with explicit methods.

Formatted input

The extraction operator >>:

- is predefined for all built-in types and strings.
- can be implemented for [user-defined data types].
 (https://www.educative.io/collection/page/10370001/5128982204776448/59
 28180936343552)
- can be configured by format specifiers.

```
#include <iostream>
//...
int a, b;
std::cout << "Two natural numbers: " << std::endl;
std::cin >> a >> b; // < 2000 11>
```

std::cout << "a: " << a << " b: " << b;

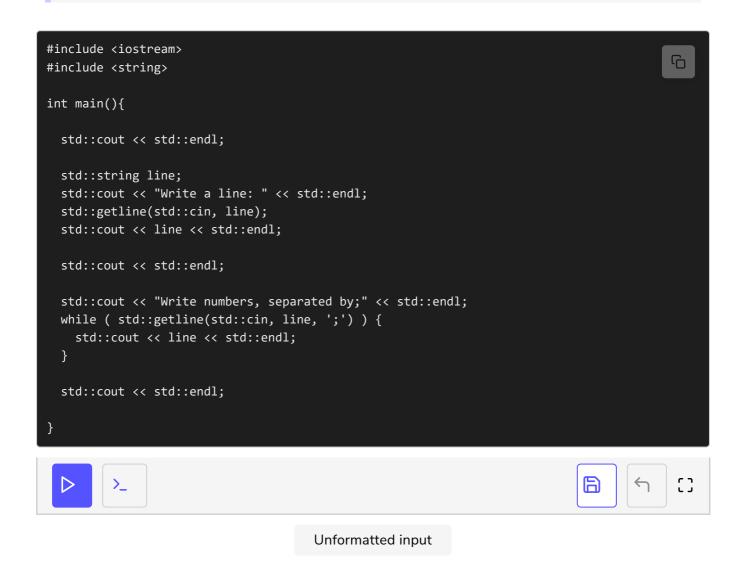
Unformatted input

There are many methods that can be used for the unformatted input from an input stream is.

Method	Description
is.get(ch)	Reads one character into ch.
<pre>is.get(buf, num)</pre>	Reads, at most, num characters into the buffer buf.
<pre>is.getline(buf, num[, delim])</pre>	Reads, at most, num characters into the buffer buf. Optionally uses the line-delimiter delim (default \n).
is.gcount()	Returns the number of characters that were last extracted from is by an unformatted operation.
<pre>is.ignore(streamsize sz= 1, int delim= end-of-file)</pre>	Ignores sz characters until delim.
is.peek()	Gets one character from is without consuming it.
<pre>is.unget()</pre>	Pushes the last read character back to is.
is.putback(ch)	Pushes the character ch onto the stream is.

std::string has a getline function

The getline function of std::string has a big advantage above the getline function of the istream. The std::string automatically takes care of its memory. On the contrary, we have to reserve the memory for the buffer buf in the is.get(buf, num) function.



Output

We can push characters with the insert operator << onto the output stream.

The insert operator <<

- is predefined for all built-in types and strings.
- can be implemented for [user-defined data types].

 (https://www.educative.io/collection/page/10370001/5128982204776448/59
 28180936343552)
- can be adjusted by format specifiers.

Further information

- getline function
- user-defined data types

Let's solve an exercise in the next lesson.