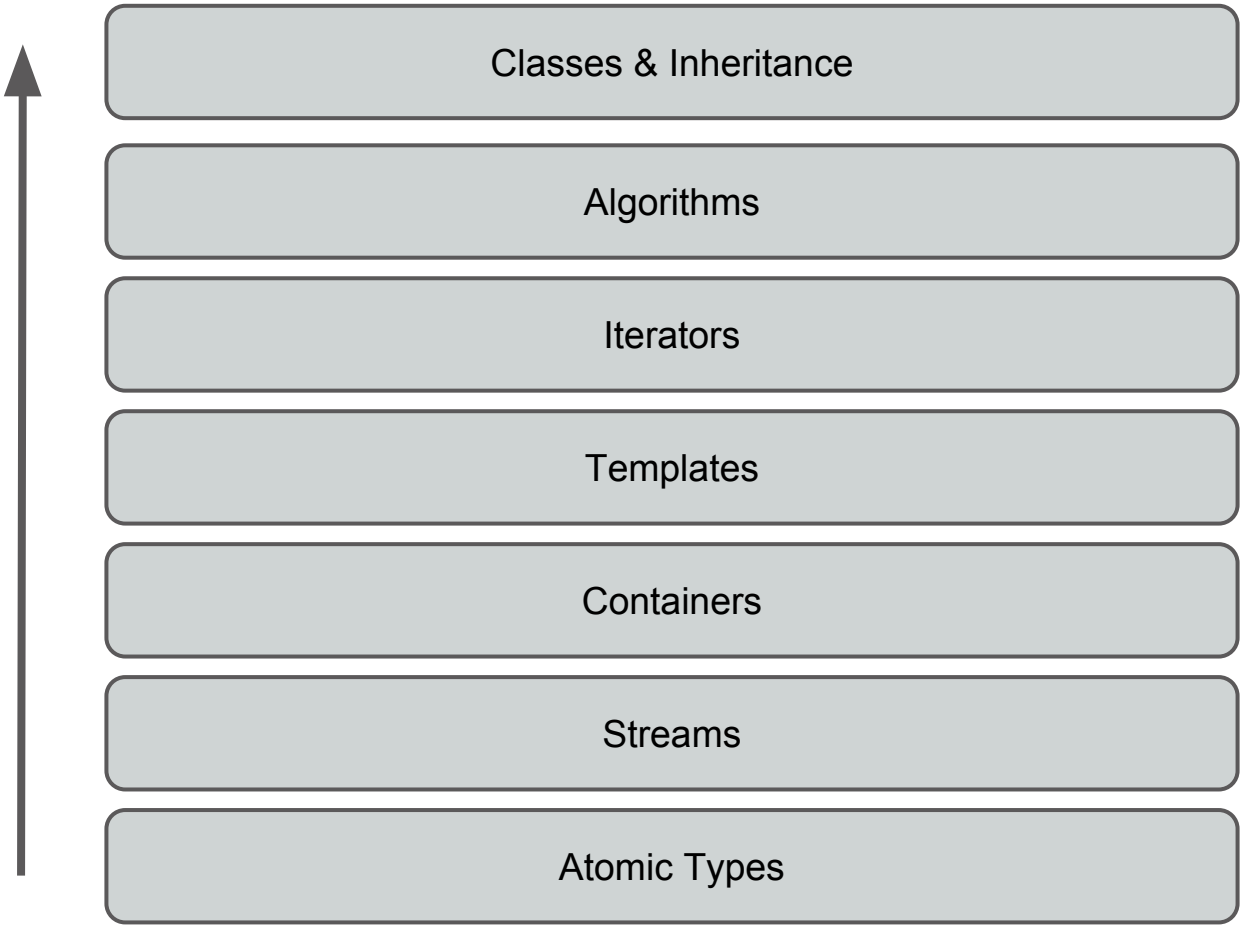
Streams

Cristian Cibils
(ccibils@stanford.edu)

Prelude 1: Roadmap!

We will follow the design of C++!

The Design of C++



Prelude 2: Code Snippets

I often present code snippets like:

```
cout << (5 * 4) + 2 << endl;
```

Prelude 2: Code Snippets

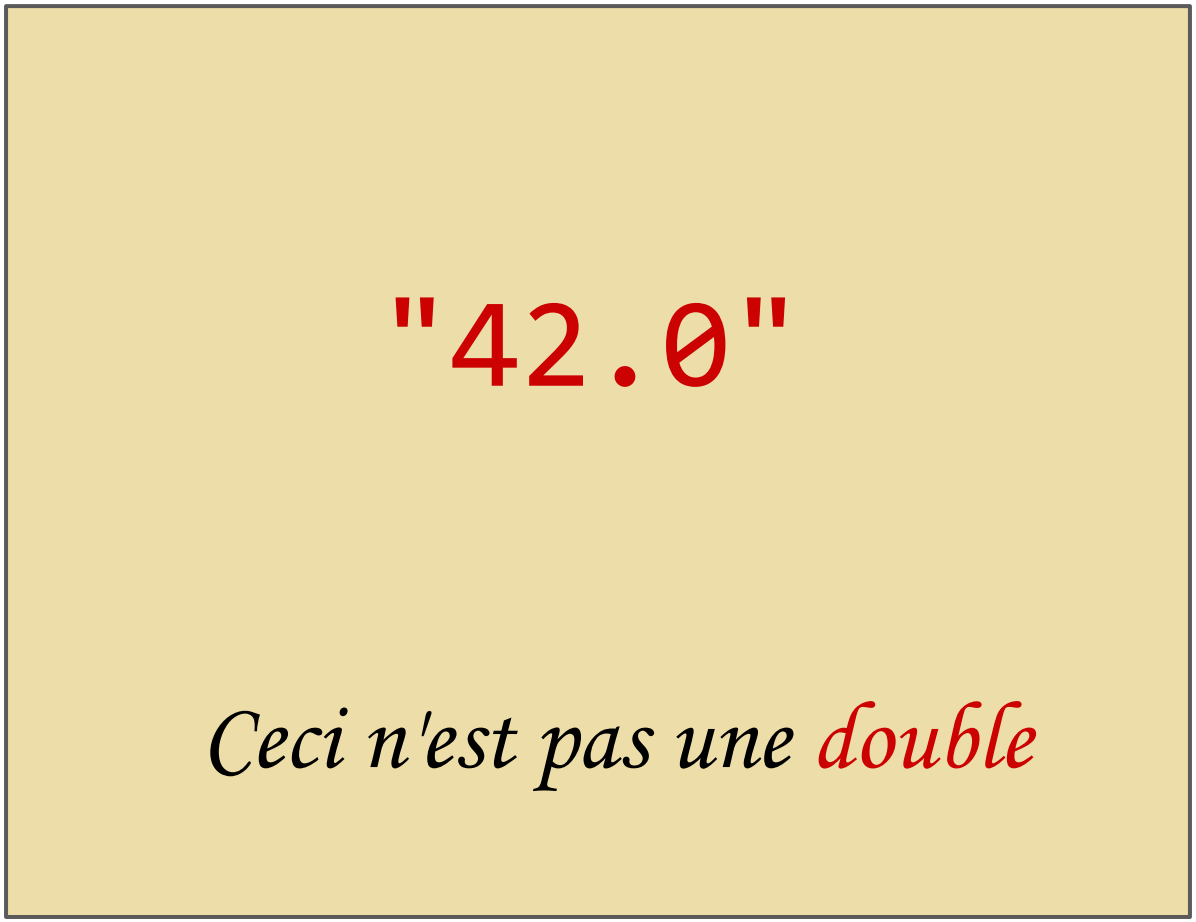
When you see a code snippet like that, think:

```
#include <iostream>
using namespace std;

int main() {
   cout << (5 * 4) + 2 << endl;
   return 0;
}
```

The Ideas Behind Streams

The Ideas Behind Streams



The Ideas Behind Streams

In C++, what is the difference between

```
double number = 42.0;
```

and

```
string number = “42.0”;
```

The Ideas Behind Streams

This won't work:

```
//print double a number
void printDouble(string s) {
    cout << s * 2 << endl;
}
```

The Ideas Behind Streams

Will this?

```
//print a number appended with 4
void append4(int n) {
   cout << n + “4” << endl;
}
```

The Ideas Behind Streams

- Input from user is in text form (`string`)
- Output to screen is in text form (`string`)
- Computation, however, needs to be done in numeric form (`int`, `double`, etc)

"Designing and implementing a general input/output facility for a programming language is notoriously difficult" --Bjarne Stroustrup

The Ideas Behind Streams

Streams allow a C++ programmer to convert between the string representation of data, and the data itself.

What is a Stream?

- a **stream** is an object that can send and receive data
- You have already been using streams: namely **cout**
- Many different kinds of streams

Hello World in C++

```
#include <iostream>
using namespace std;

int main() {
   cout << "Hello World!" << endl;
   return 0;
}
```

Hello World in C++

```
#include <iostream>
using namespace std;

//Sends the string “Hello World!”
//to the output stream (ostream) cout
int main() {
   cout << "Hello World!" << endl;
   return 0;
}
```

Output Streams

- Any stream which you can only *send* data to, like cout, is called an output stream, or `ostream`
- Send data using the string insertion operator: `<<`
- Converts data to a string **and** sends to a stream

Output Streams

- You can use an `ostream` for more than just printing data to a console
- You can also print data to a file using an `ofstream`

Output Streams

Output Stream Example
(OutputStreams.pro)

Time Out

- Office Hours:
  - Tuesdays and Thursdays after class.
  - Right here!

Input Streams

Is this familiar?

```
int x;
cin >> x;
```

Input Streams

- Any stream which can only give you data, like cin, is called an input stream, or `istream`
- Send data using the string extraction operator: `>>`
- Gets data from the stream **and** converts it into the appropriate type

Input Streams

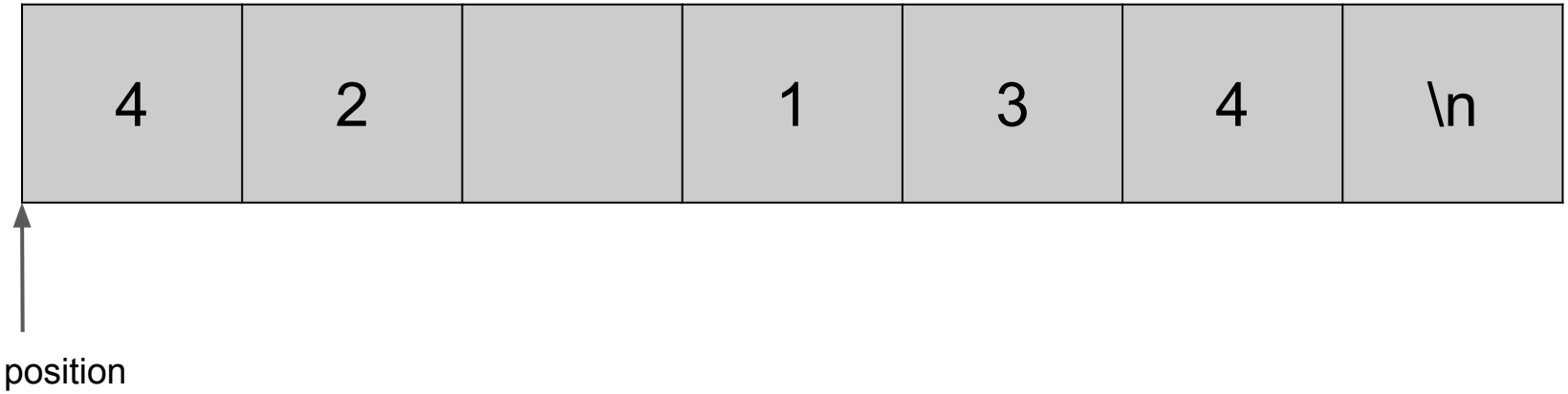
- Just like with an `ostream`, an `istream` can be used or more than just console IO
- You can also read data from a file using an `ifstream`

Input Streams

Input Stream Example
(InputStreams.pro)

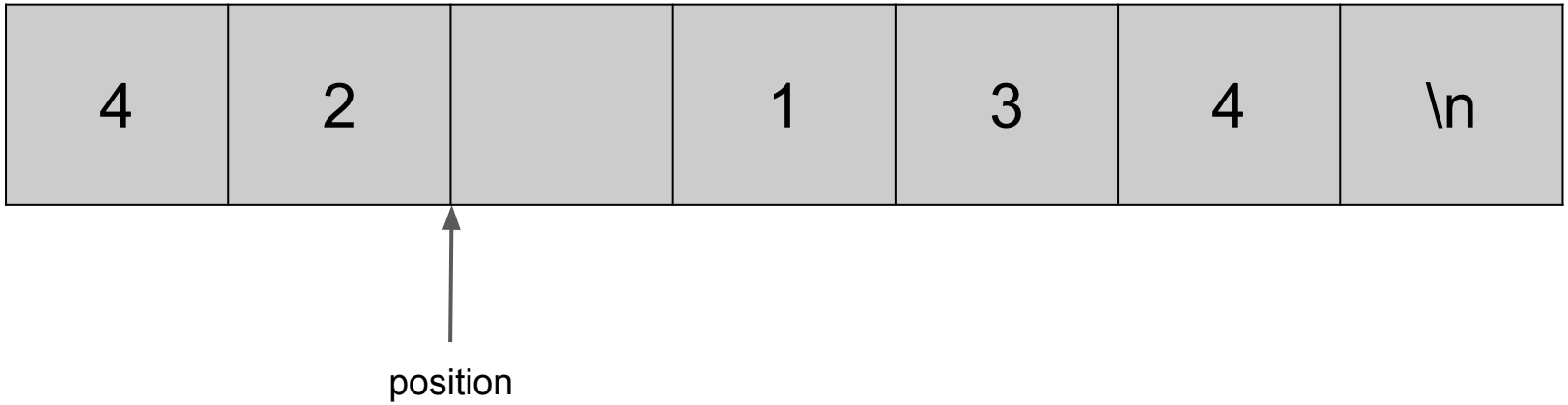
Reading Data from a File

To understand an `istream`, think of it as sequence of characters



Reading Data from a File

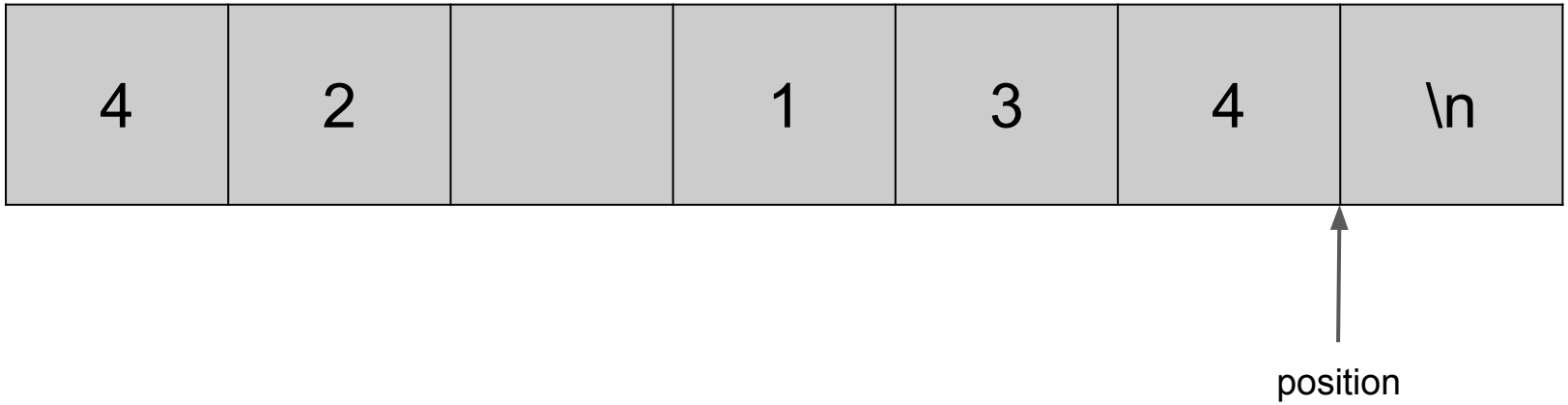
Extracting an integer will read as many characters as possible from the stream




```
int value;
istream >> value //value == 42
```

Reading Data from a File

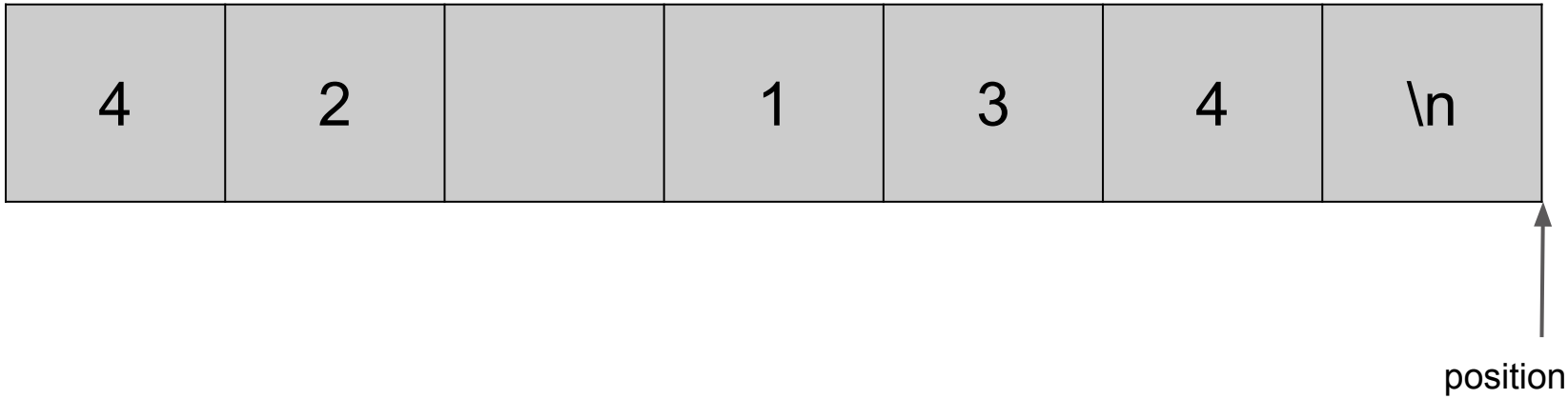
Extracting again will skip over any whitespace when reading the next integer




```
int value;
istream >> value //value == 134
```

Reading Data from a File

When no more data can be read, the **fail bit** will be set to `true`




```
int value;
istream >> value //value == ??
```

Input Streams

More Input Stream Examples
(InputStreams.pro)

Reading Data from a File

- There are some quirks with extracting a string from a stream
- Reading into a string using >> will only read a single word, not the whole line
- To read a whole line use getline

```
getline(stream, string line);
```

Reading Data from a File

- To re-read a file, you must close it, clear it, and reopen it

```
input.close();
input.clear();
input.open("filename");
```

Input Streams

Most Input Stream Examples
(InputStreams.pro)

Reading Data from a File

- There are some more subtle details to file reading that we will examine on Tuesday
- As a preview: What happens when you read into the wrong type?

R	a	p	t	o	r	\n

```
int x;
input >> x;
```

One Last Stream: Stringstream

- There is another type of stream worth mentioning: the stringstream
- Unlike every other stream we have looked at, stringstreams don’t send data anywhere
- Useful for converting between types

One Last Stream: Stringstream

Stringstream Examples

(StringStream.pro)