The Little CPPler

idea: at some point in the code explain why is the book written in English and no any other language

Prerequisites

What to expect from this book

A phrase telling them what to do with the book

Tools

Tools neccesary to follow the book (for everything to work), note: the source code can be found in a repo, thought

Using the tools

Ask to open a file with a text editor

Extend explanation on how to compile

ask to modify the file and see what happens

Given the file `000-hello-world.cc`:

```
#include <iostream>

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

compile the file into a executable with either `clang++` or `g++`:

add std option to make it C++11 compilant

```
clang++ 000-hello-world.cc -o 000-hello-world.exe
g++ 000-hello-world.cc -o 000-hello-world.exe
```

and run it with:

```
./000-hello-world.exe
```

the output of the program in terminal will be:

```
Hello World!
```

Content

How to read this book

Yeah, you should learn first how to read this book! :P

Explain how the book is supposed to be read (try to answer question with answer covered, then reveal answer and try understand what is it doing)

Basics

Getting to know C++(11)

Add interludes asking whoever is reading to pause for a while to recover from so much info

---

000.

What do you think the following code will output after compiling and running it?

```cpp
#include <iostream>

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

The output is:

```
Hello World!
```

001.

Now, what if you compile this other file:

```cpp
#include <iostream>

int main()
{
  std::cout << "Hello World!" << std::endl;
  std::cout << "I'm a program example and I'm "
            << "in English."
            << std::endl;
  return 0;
}
```

The output is:

```
Hello World!
I'm a program example and I'm in English.
```

Pay close attention to the output, there are three sentences surrounded by quotation marks ("), but there are only two lines in the output. Why?

002.

If we change our example slightly (notice the semicolon (;)) what do you think it will happen?

```
#include <iostream>

int main()
{
  std::cout << "Hello World!" << std::endl;
  std::cout << "I'm a program example and I'm ";
            << "in English."
            << std::endl;
  return 0;
}
```

Well, it doesn't compiles! We get an error similar to:

```
test.cc:7:13: error: expected expression
            << "in English."
            ^
```

It is telling us that it was expecting something (a `std::cout` for example) before `<<`.
Try removing or adding random characters (anywhere) to the example and you will find that the compiler just admits a certain arrangement of characters and not much more. But, why? Well, the compiler just understands the grammar of C++ as we just understand the grammar of our human languages. Going a little further with the analogy, we can understand the grammar of any human language (its parts (verbs, prepositions, …) and how are they connected) but we can only understand the meaning (semantics) of those languages we have studied (or our mother tongues).

003.

Does the following program compiles. If yes, what is its output?

```
#include <iostream>

int main()
{
  std::cout << "Hello World!" << std::endl;
  std::cout << "I'm a program"
            << std::endl
            << "example and I'm"
            << std::endl
            << "in English."
            << std::endl;
  return 0;
}
```

Yep, it in fact compiles, and its output is:

```
Hello World!
I'm a program
example and I'm
in English.
```

Notice how `std::endl << "hello"` puts text in a new line, that's in fact its whole job.

004.

Well that's getting boring. What if we try something different for a change. What is the otput of this program:

```
#include <iostream>

int main()
{
  std::cout << "Adding two numbers: "
            << 2 + 3
            << std::endl;
  return 0;
}
```

Nice[†]

```
Adding two numbers: 5
```

[†]this is a footnote, read all of them, they may tell you little things that the main text won't.

005.

Let's try something a little more complex[†]

```
std::cout
  << "A simple operation between " << 3
  << " " << 5 << " " << 20 << ": "
  << (3+5)*20 << std::endl;
```

```
A simple operation between 3 5 20: 160
```

[†]Here you can see only a snippet of the whole code. The complete code the snippet represents can be found in the source accompaning this book.

From now on, all code will be given on snippets for simplicity but remember that they are that, snippets, uncomplete pieces of code that need your help to get complete.