

Hello, World!

Chapter 2

“Programming is learned by writing programs”

Brian Kernighan

Writing Programs

Giving directions is an example of programming

Think about giving someone directions to Lane Stadium.

What would we have to tell them?

How much detail would you have to give?

What if the “person” was a computer?

“Computers are *really* dumb.” p44

Classic First Program

//this program outputs the message “Hello, World!” to the monitor

```
#include <iostream>
```

```
int main()
```

```
{
```

```
    std::cout << “Hello, World!\n”;
```

```
    //output “Hello, World!”
```

```
    return 0;
```

```
}
```

What is this?

This is a set of text instructions that tell the computer what to do.

`cout << "Hello, World!";` is the line that actually tells the computer to print the characters Hello, World! to the screen.

`cout` (pronounced see-out) stands for **character output**.

Really?

The line with the `//` is a comment that is for the human reader, looking at the program.

It's ignored by the compiler, or not translated into machine code.

The `return 0;` is what the short program gives back as an ending value.

A brief comment on commenting

Comments are written to describe what the program does, or is supposed to do.

The person who is most likely to benefit from the comments is you.

The comments tell you what the program was supposed to do. There are times when you'll come back and look at code you previously wrote and not remember what it was supposed to do.

#include

The first line is a #include line.

Lines that start with a # are preprocessor statements.

#include lines allow you to use pre-written code in your code.

Main

Every program in C++ must have one and only one main function.

The main function returns or gives back an integer, or an int.

So we have `return 0;` as our last line.

We can return other integers, but 0 indicates success.

Functions

“A function is basically a named sequence of instructions for the computer to execute in the order in which they are written.” p47

A function has 4 parts:

1. A return type,
2. A name,
3. A parameter list enclosed in parenthesis,
4. A function body.

Compilation

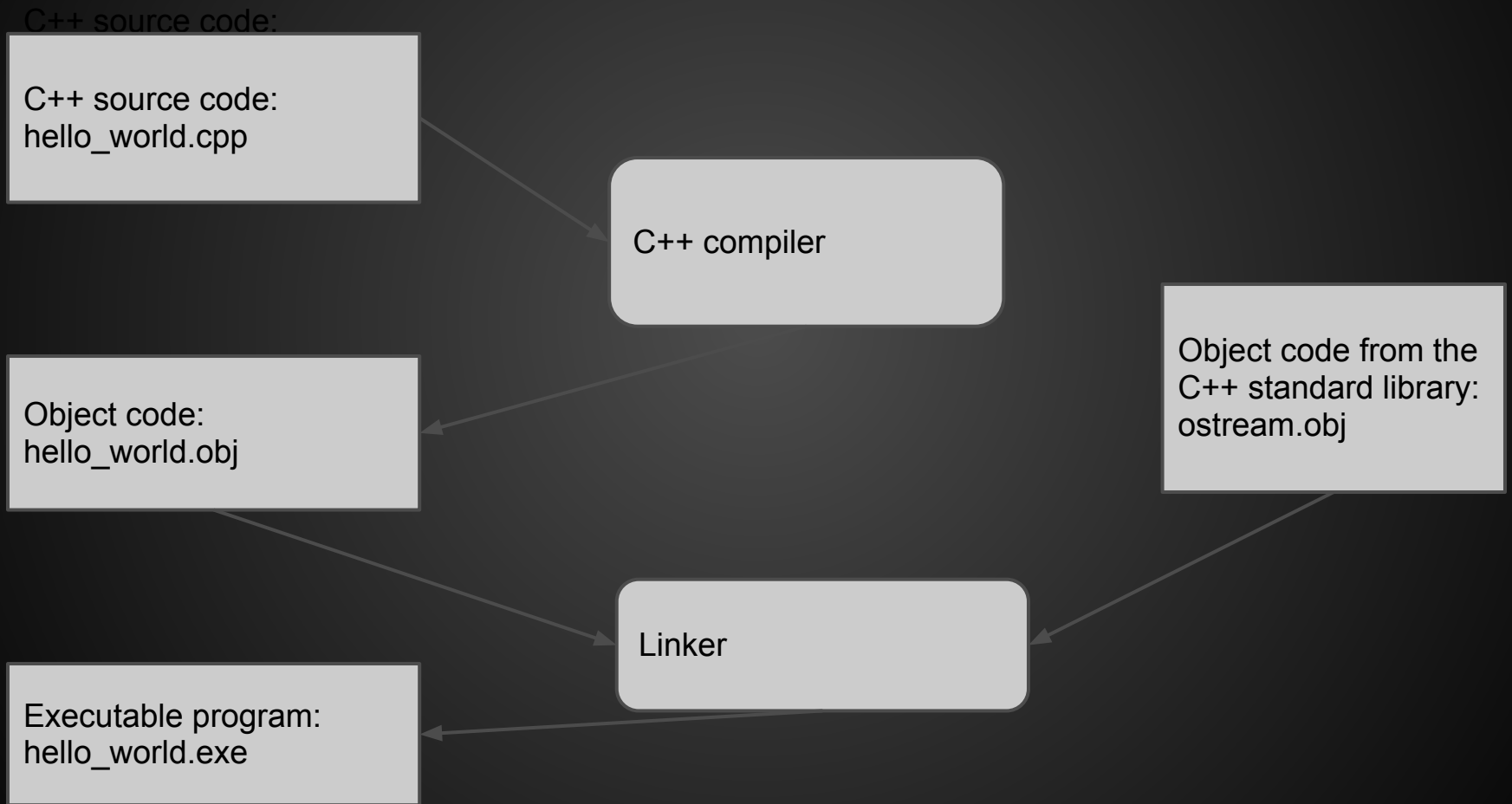
Before we can “run” our program, we must compile it.

A compiler is just a program that takes your text file and produces another file, an object file.

“The compiler is your friend; possibly, the compiler is the best friend you have when you program.” p50

Linking

Then a linker, another program, takes that output file, the object file, and the other needed files and puts them all together into an executable.



Errors

Compile time errors are found by the compiler.

These are sometimes called syntax errors.

Relatively easy to find and fix.

Linking errors are found by the linker and are harder to find and fix.

Logical errors or run-time errors are found during running or testing.

They are the hardest to find and fix.