

Introduction to C++ Part 1

Chapter 1.3-1.4

Slidedeck #2
CMPT 130

© Brian Fraser



Topics

- 1) What does a **simple C++ program** look like?
- 2) How can we **output text** to the screen?
- 3) What kind of **errors** will we see?

Hello World!

A simple C++ program.

Simple C++ Program

C++ Source Code

```
// A simple C++ program.  
#include <print>  
using namespace std;  
  
int main() {  
    println("Hello world");  
    return 0;  
}
```

Output

Hello world

Simple C++ Program

```
// A simple C++ program.  
#include <print>  
using namespace std;  
  
int main() {  
    println("Hello world");  
    return 0;  
}
```

Comments:

All text on a line after
a *//* is a comment.

These are notes to
the programmer;..

Simple C++ Program

```
// A simple C++ program.  
#include <print>  
using namespace std;  
  
int main() {  
    println("Hello world");  
    return 0;  
}
```

#include:

Tells the compiler that we may print to the screen in our program.

This is called a ..

The *header* file *print* is included for our use.

Simple C++ Program

```
// A simple C++ program.  
#include <print>  
using namespace std;  
  
int main() {  
    println("Hello world");  
    return 0;  
}
```

using namespace:

All **identifiers** (such as variable and function names) are inside a **namespace**.

Basically, this states that we want to use identifiers in the **std** namespace.

..

Simple C++ Program

```
// A simple C++ program.  
#include <print>  
using namespace std;
```

```
int main() {  
    println("Hello world");  
    return 0;  
}
```

int main() { ... }:

Creates the **main()** function.

The **main()** function is..

Functions are named
collection of statements.

Note: C++ is case sensitive!
main() is different than Main() or MAIN()!

Simple C++ Program

```
// A simple C++ program.  
#include <print>  
using namespace std;  
  
int main() {  
    println("Hello world");  
    return 0;  
}
```

int is the **return type**.

In this case, an integer.

It is the type of information the program "returns" to the OS.

main is the name of our function.

Each program we create must..

The **()** indicates this is...

The **{ ... }** indicates a **block**.

In this case, a block of statements associated with the **main()** function.

Simple C++ Program

```
// A simple C++ program.  
#include <print>  
using namespace std;  
  
int main() {  
    println("Hello world");  
    return 0;  
}
```

println...

The “ln” (LN) means Line Feed:
move to next line after printing

Completed statements end with a semicolon.
We will later get a better feel for this.
For now, just concentrate on the parts of the program.

Simple C++ Program

```
// A simple C++ program.  
#include <print>  
using namespace std;  
  
int main() {  
    println("Hello world");  
    return 0;  
}
```

return:

The **return** statement in the **main()** function returns a value to the operating system.

Returning 0 to the OS indicates success (by convention).

..

Review

- 1) What C++ statement prints "I love programming" to the screen?
- 2) What part of a C++ program is first to be executed?
- 3) What is wrong with this C++ statement?
`PrintLN("Hello!");`

Tools

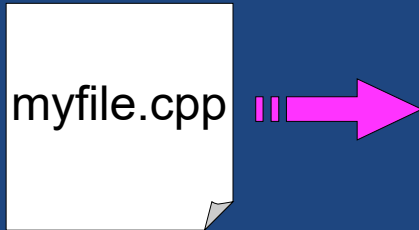
How to build
an executable.



Build Process

- We write C++ code; computer runs machine code.

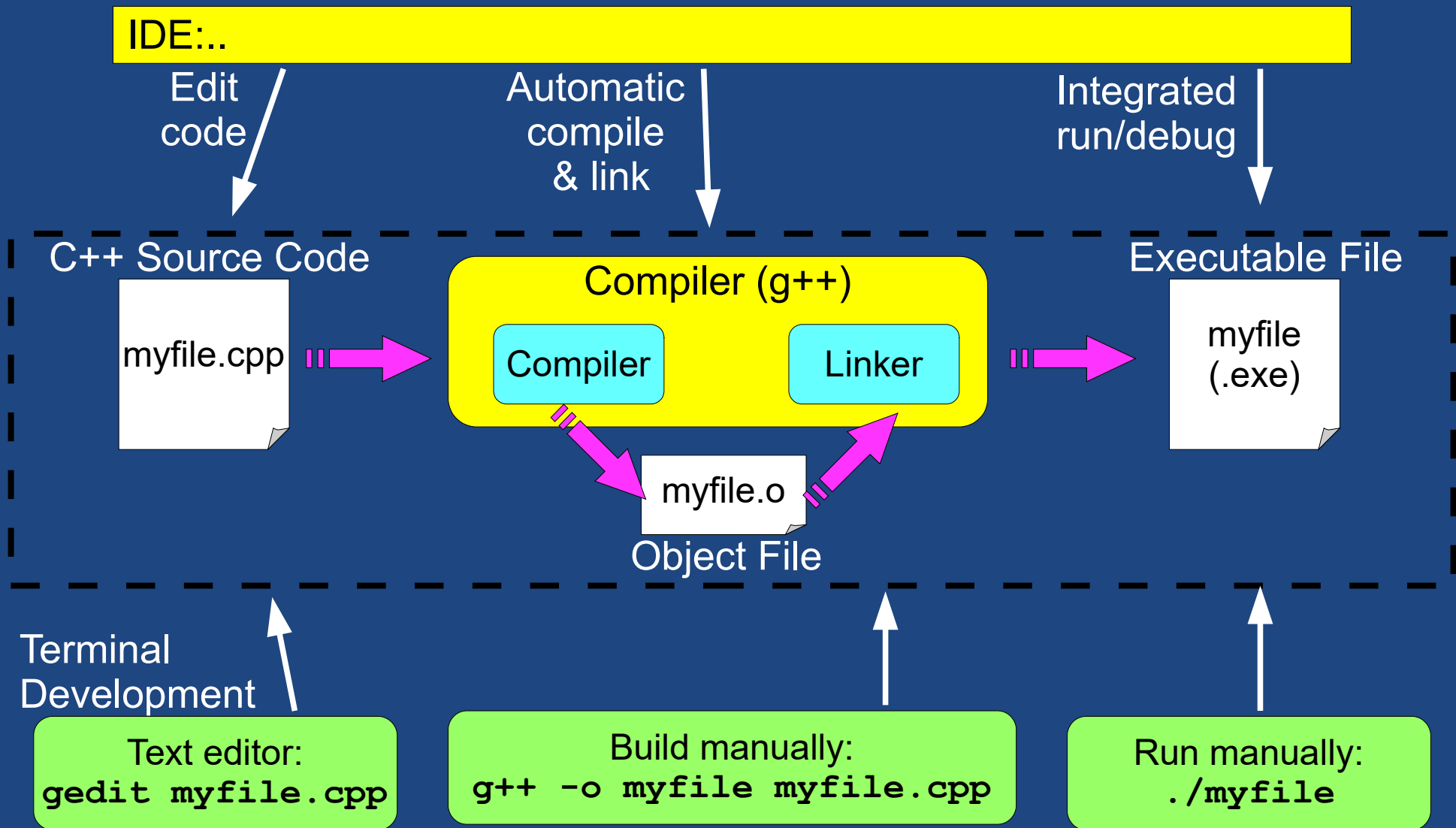
C++ Source Code



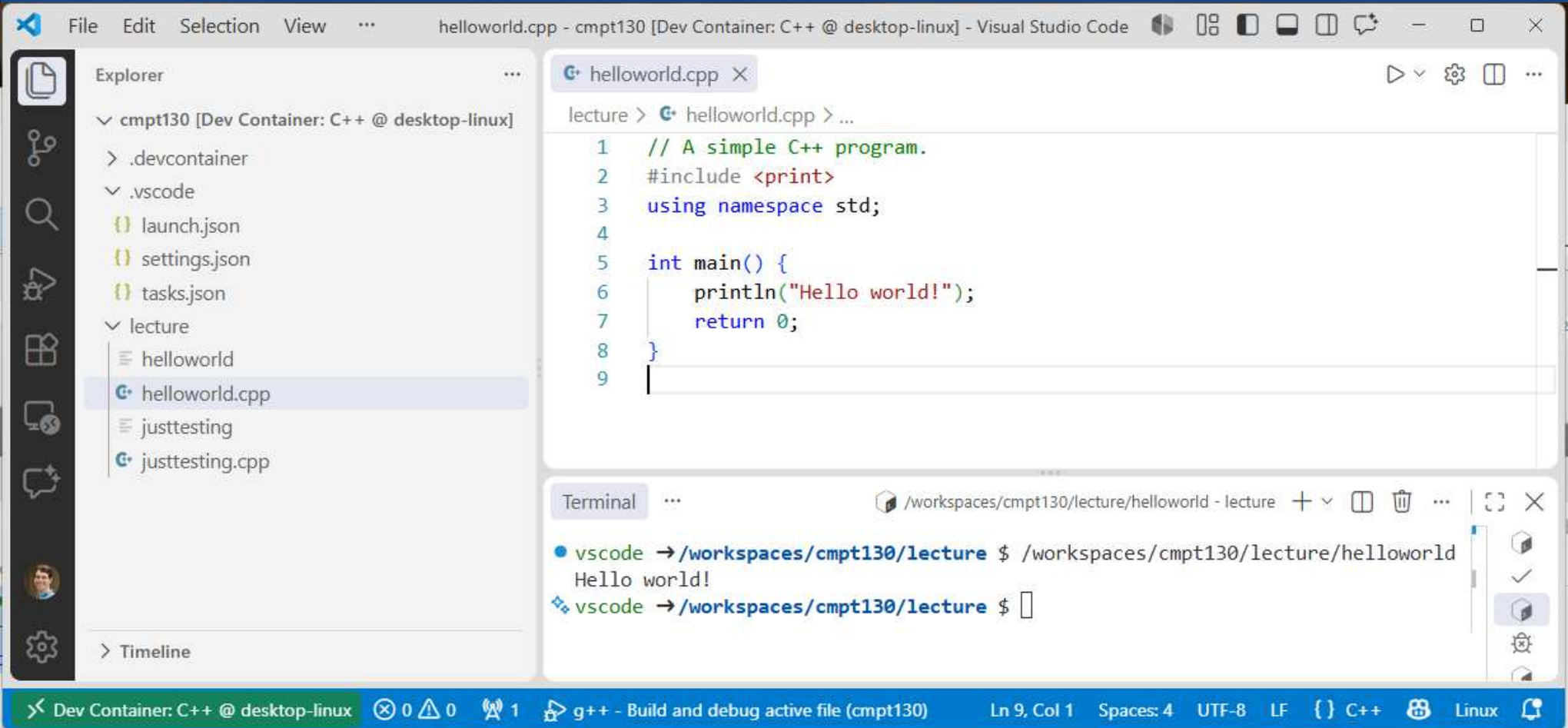
Executable File

- Build...
- Tool Options
 - IDE...
All tasks done through graphical user interface (UI)
 - Terminal Development:
All tasks done through a command prompt.

Tools Options



Visual Studio Code



Summary

- How to create a C++ program: **Hello World**
- **Output** to the console with **print/println**.
- **Compile** C++ code to machine code.

Introduction to C++ Part 2

Chapter 1.3-1.4

Slidedeck #2

CMPT 130

© Brian Fraser



Controlling the Output

println vs print()

- Can print to screen with either `print()` or `println()`:

```
// print vs println
#include <print>
using namespace std;

int main() {
    print("Hello");
    print("World");
    println("!");

    println("Hello World!");
    return 0;
}
```

Notice all the strings are run together, even though they are from separate statements.

HelloWorld!
Hello World!

Special Characters

- Escape Sequences:

..

- New line: "One \n on \n top"
 - Tabs (line up): "Age: \t23"
 - A \ character: "Up \\ down"
 - A ' character: "\"I'm lovin\" programming!"
 - A " character: "I said, \"Yes!\" too"
- The escape sequence must be inside a string.

Escape Sequence Example

```
// Demonstrate escape sequences
#include <print>
using namespace std;

int main () {
    println("Movie Lineup");
    println("7:30\tSpace Balls");
    println("10:40\tIt\'s a Wonderful Life");
    println("12:30\tGone with the Wind");
    println("*\n*\n*");
    println("He'll say, \"They\'re great!\"");
    return 0;
}
```

```
Movie Lineup
7:30   Space Balls
10:40  It's a Wonderful Life
12:30  Gone with the Wind
*
*
*
He'll say, "They're great!"
```

Spot the Mistakes

Each line demonstrates a different problem!

```
\\ Spot some common mistakes.
include <print>

int main
[
    print("Welcome!");
    print("*****")
    print("I say, /"Yes!/"");
    print( Little things matter. );
    print('Thanks for playing!');
    return 1;
]
```

Spot the Mistakes

Each line demonstrates a different problem!

Program with Errors

```
\\ Spot some common mistakes.
include <print>

int main
[
    print("Welcome!");
    print("*****")
    print("I say, /"Yes!/"");
    print( Little things matter. );
    print('Thanks for playing!');
    return 1;
]
```

Corrected Program

```
// Spot some common mistakes.
#include <print>
using namespace std;

int main()
{
    println("Welcome!");
    println("*****");
    println("I say, \"Yes!\"");
    println("Little things matter.");
    println("Thanks for playing!");
    return 0;
}
```

Review

- 1) Write one or more C++ statements which output the following (including tabs, and line-feeds):

Name:	"Brian"
Fav-Colour:	Green

The `cout` Object

cout

- C++ can also use cout to print to screen.
 - A very common way, but we'll use the easier print/prinln (added in 2023).
- cout (Pronounced "C Out", not "kout")
 - ..
 - Think of it as character out, or console out.

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

int main() {
    cout << "Hello world!" << endl;
    cout << "Isn't this amazing\n";
    cout << "I can " << "print " << "multiple things on a line" << endl;
    return 0;
}
```

```
Hello world!
Isn't this amazing
I can print multiple things on a line
```

cout

- **cout** is a **stream object**:
 - It operates on a **stream** (sequence) of characters.
- **<<** is the **stream-insertion operator** (or “put-to”):
 - Use it to push text into **cout**
cout << "Wow! Programming is fun!";
 - Think of **<<** as an arrow point to the left:



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

int main() {
    cout << "Hello world!" << endl;
    cout << "Isn't this amazing\n";
    cout << "I can " << "print " << "multiple things on a line" << endl;
    return 0;
}
```

```
Hello world!
Isn't this amazing
I can print multiple things on a line
```

println vs cout

- On assignments and labs, use `println()` and `print()`
- Why use `println`?
 - We'll use `print()` and `println()` because..
 - You need to know how to read code that uses `cout` because there is a lot of it in the world!
 - Some slides will still use `cout`; that's ok.

To err is human,
but to really foul things up
you need a computer.

Paul Ehrlich



1982: Bug in software
controlling Soviet pipeline
causes largest man-made non-
nuclear explosion in history.

Errors

- Compile Error
 -
 - Syntax errors, such as forgetting a ;
 - Semantic errors, such as invalid type casting.
- Run-time Error
 - Errors causing...
such as an un-checked divide by zero (exceptions).
- Logical Error
 -
 - Caused by programmer error (bug).

Debugging

- Most (all?) large programs have bugs.
 - You'll spend a large amount of time debugging!
- QA is Quality Assurance
 - The task of showing the program is..
 - Cannot reasonably **prove** that there are no bugs:
 - Can show it works for..
Ex: square root of 16?
 - Can show it works for..
Ex: square root of 0? -1? 4 billion?

Summary

- Escape Sequence: `\n`, `\t`, `\\`, `\'`, `\"`
- May see `cout` to print to screen.
- 3 types of errors:
 - Compile, run-time, logical.