Introduction to C++
Chapter 1.3-1.4

Slidedeck #2
CMPT 130
© Brian Fraser
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.
// A simple C++ program.
#include <iostream>
using namespace std;

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

int main () {
    cout << "Hello world";
    return 0;
}

Comments:
All text on a line after a // is a comment.

These are notes to the programmer;..
// A simple C++ program.
#include <iostream>
using namespace std;

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

int main () {
    cout << "Hello world";
    return 0;
}

using namespace std;

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.

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

int main () {
    cout << "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()!
// A simple C++ program.
#include <iostream>
using namespace std;

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

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

int main () {
    cout << "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?
   cOut >> "Hello!"
Tools

How to build an executable.
Build Process

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

C++ Source Code

myfile.cpp

Executable File

myfile (.exe)

- Build:

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

IDE:
- Edit code
- Automatic compile & link
- Integrated run/debug

C++ Source Code
- myfile.cpp

Compiler (g++)
- Compiler
- Linker
- myfile.o

Object File

Executable File
- myfile (.exe)

Terminal Development
- Text editor: gedit myfile.cpp
- Build manually: g++ -o myfile myfile.cpp
- Run manually: ./myfile
```cpp
// A simple C++ program.
#include <iostream>
using namespace std;

int main () {
    cout << "Hello world";
    return 0;
}
```
The cout Object
cout

- cout (Pronounced “C Out”, not “kout”)
  - Think of it as character out, or console out.

- cout is a stream object:
  - It operates on a stream (sequence) of characters.

- << is the stream-insertion operator:
  - Use it to push text into cout
    cout << "Wow! Programming is fun!";
  - Think of << as an arrow point to the left:
    ![Diagram](image-url)
You can send multiple different strings to `cout`:

```cpp
// Displaying multiple strings.
#include <iostream>
using namespace std;

int main () {
    cout << "Programming is " << "great fun ";
    cout << "all the time!";

    return 0;
}
```

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

• What is the problem with the following?

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

int main () {
    // Demonstrate a common problem
    cout << "My favourite numbers are: ";
    cout << "0";
    cout << "42";
    cout << "73";
    return 0;
}
```
Line Feeds

- Can put line feeds in with either:
  - **End Line Stream Manipulator**: `endl`
    ```
    cout << "First line." << endl;
    cout << "Second." << endl << "Third.";
    ```
  - **New Line Character**: `\n`
    ```
    cout << "First line.\n";
    cout << "Second.\n" << "Third."
    ```
Special Characters

• **Escape Sequences:**
  
  - New line: "One \n on \n top"
  - Tabs (line up): "Age: \t"
  - A \ character: "Up \ down"
  - A ' character: "I'm lovin' programming!"
  - A " character: "I said, "Yes!" too"

• Note that the escape sequence must be inside a string, whereas `endl` must **not** be in the string.
// Demonstrate escape sequences and endl
#include <iostream>
using namespace std;

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

// Show some common mistakes.
#include <iostream>
using namespace std;

int main () {
    // Spot the mistakes:
    cout << "C++ is fun! endl";
    cout << "Computers are awesome!" << \n;
    cout << "Amazing stuff!\n";
    cout << "I say "Yeah!"") >> endl;
    return 0;
}
1) Write one or more C++ statements which output the following (including tabs, and line-feeds):

Name: "Brian"
Fav-Colour: Green
Errors

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

Paul Ehrlich

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

- Simple program: "Hello world!"
- Output to the console with `cout`.
  - `cout << "One " << "Two";`
  - `cout << "With 2 line feeds\n" << endl;`
- Compile C++ code to machine code.
- Escape Sequence: `\n, \t, \, \', \``
- 3 types of errors:
  - Compile, run-time, logical.