CSCI 200: Foundational Programming Concepts & Design Lecture 03

Random

Command Line Interface

Makefiles

Have VS Code & iClicker open

Download Command Line Cheat Sheet & Starter Code for today

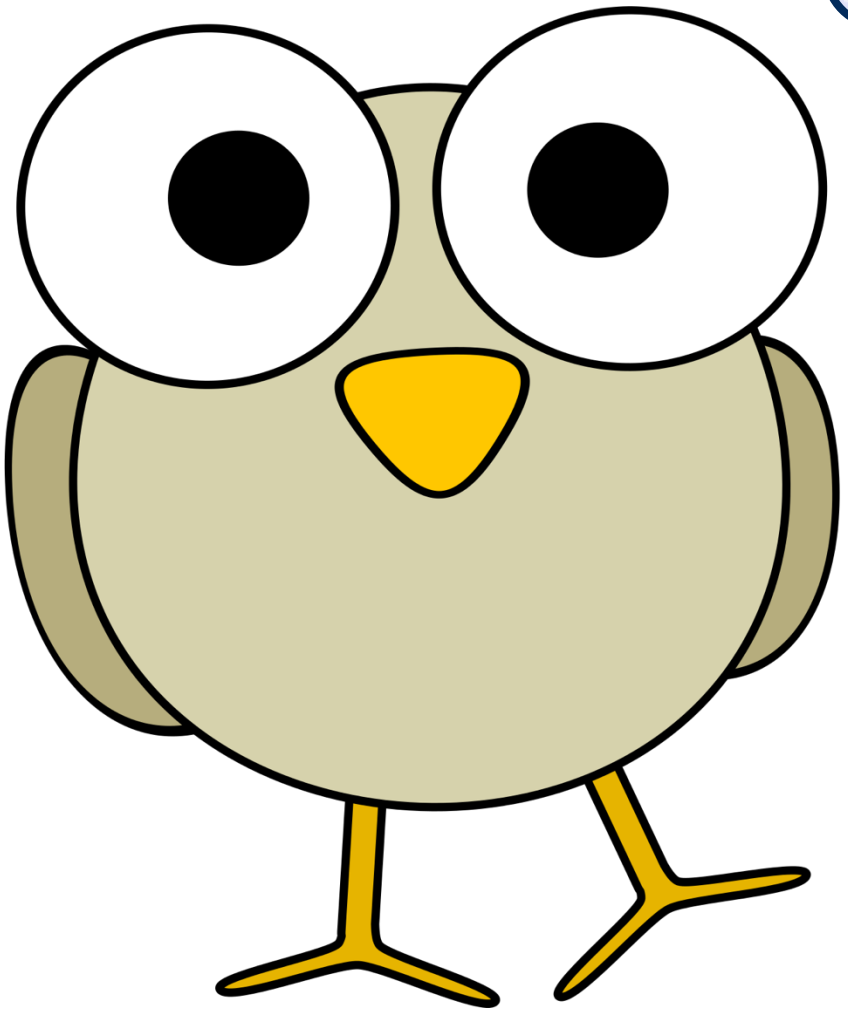
Note on Precision

- For floating point values, the magnitude affects significance
- If 7 digits of precision:
  - 0.000abcdefg??
  - 0.abcdefg??
  - abcd.efg???
  - abcdefg????.???
- Trailing values may not be accurate

Precision Fun Fact

Precision	Bits	Bit Breakdown	# Digits Precision	Range
Half	16	1 sign 5 exponent 10 mantissa	~3	$\pm 10^{-5}$ to ± 65504
Single **float**	32	1 sign 8 exponent 23 mantissa	~7	$\pm 10^{-38}$ to $\pm 10^{38}$
Double **double**	64	1 sign 10 exponent 53 mantissa	~16	$\pm 10^{-308}$ to $\pm 10^{308}$
Quad	128	1 sign 14 exponent 116 mantissa	~33	$\pm 10^{-4932}$ to $\pm 10^{4932}$
Oct	256	1 sign 18 exponent 237 mantissa	~71	$\pm 10^{-78913}$ to $\pm 10^{78913}$

Questions?



Learning Outcomes For Today

- List common Linux terminal commands and choose the correct commands to work with a file system via the command line.
- Describe how a computer generates a program from code.
- Describe how a computer generates random numbers.
- Write and use a Makefile.
- Discuss the advantages of using

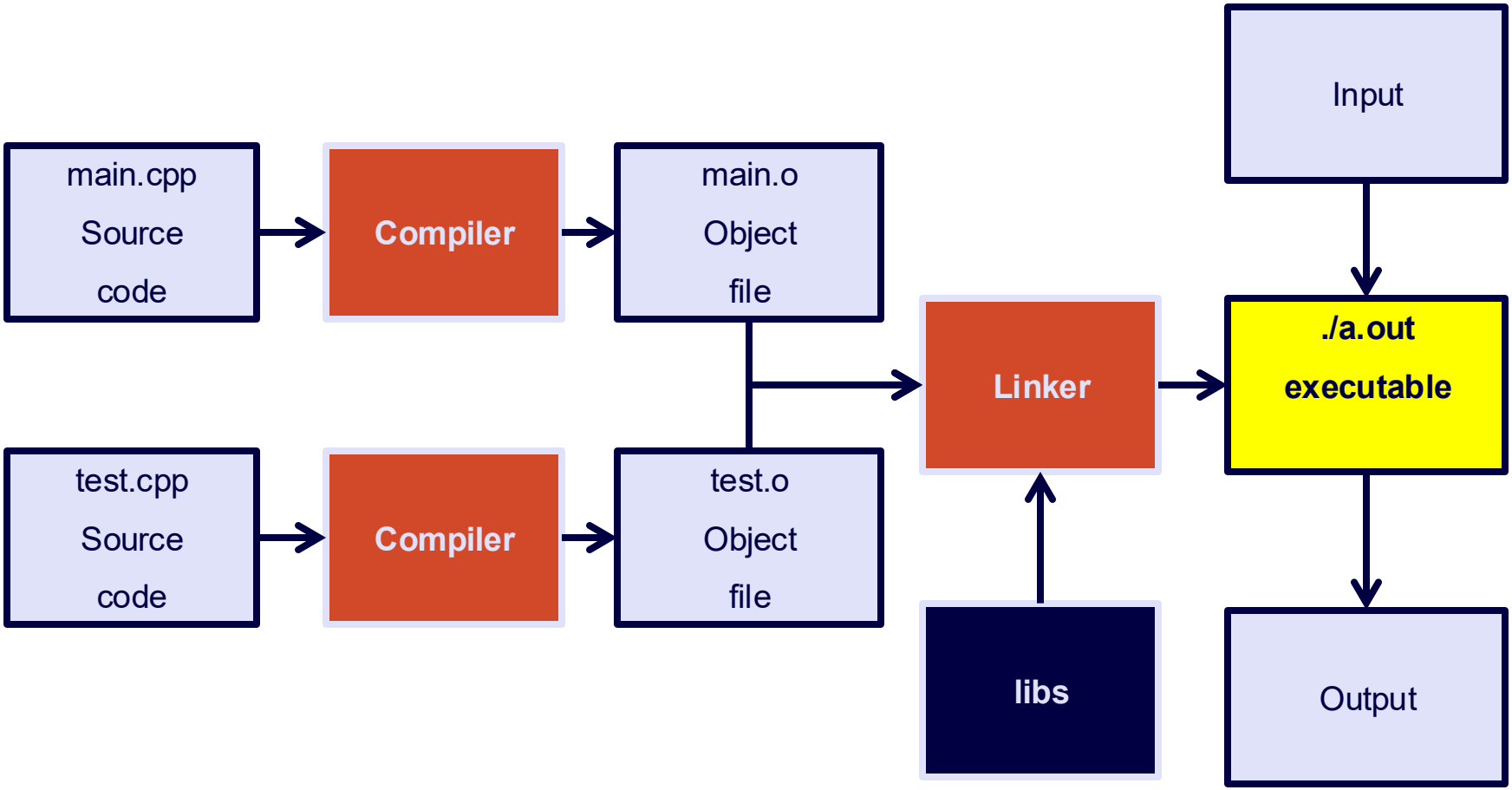
On Tap For Today

- Building a C++ Program
- Compiler Flags & Directives
- Makefiles
- Practice

On Tap For Today

- Building a C++ Program
- Compiler Flags & Directives
- Makefiles
- Practice

Compile & Link Process



Command Line Interface (CLI)

- Textual representation to move through file system and directory structure

CLI Cheat Sheet

- Directory Operations
  - Show current directory
  - List files
  - Make directory
  - Change directory
    - Go up a directory
  - Copy a directory (**NO UNDO**)
  - Move a directory^ (**NO UNDO**)
  - Remove a directory^ (**NO UNDO**)

^Slight differences between Windows / OS X for copy/move/remove directory

CLI Cheat Sheet

- File Operations
  - Create file^
  - Copy file (**NO UNDO**)
  - Move file (**NO UNDO**)
  - Remove file (**NO UNDO**)

^Slight differences between Windows / OS X for creating a file

CLI Cheat Sheet

- Program Operations
  - Build program
  - Run program^

^Slight differences between Windows / OS X when running program

Using the Command Line

- Before starting: extract `L03.zip` somewhere on your machine

- Complete the following tasks **via** the terminal:
  - Create a folder named `Lectures`
  - Inside the `Lectures` folder
    - Create a folder named `Lecture03`
    - Inside the `Lecture03` folder
      - Copy the file named `main.cpp`

```
mkdir Lectures
cd Lectures
mkdir Lecture03
cd Lecture03
cp ~/L03/main.cpp .
cp ~/L03/Makefile .
```

On Tap For Today

- Building a C++ Program
- Compiler Flags & Directives
- Makefiles
- Practice

Compiler Directives

- The C++ code also tells the compiler information

```
#include <iostream>
```

- C++ statements beginning with # are compiler directives

#include

```cpp
// main.cpp
#include <iostream>
using namespace std;

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

```cpp
// iostream
// ...
cout = ...
endl = ...
```

#include

```
// main.cpp
#include <iostream>
using namespace std;

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

```
// iostream
// ...
cout = ...
endl = ...
```

#include

```
// main.cpp
// iostream
// ...
cout = ...
endl = ...
using namespace std;


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

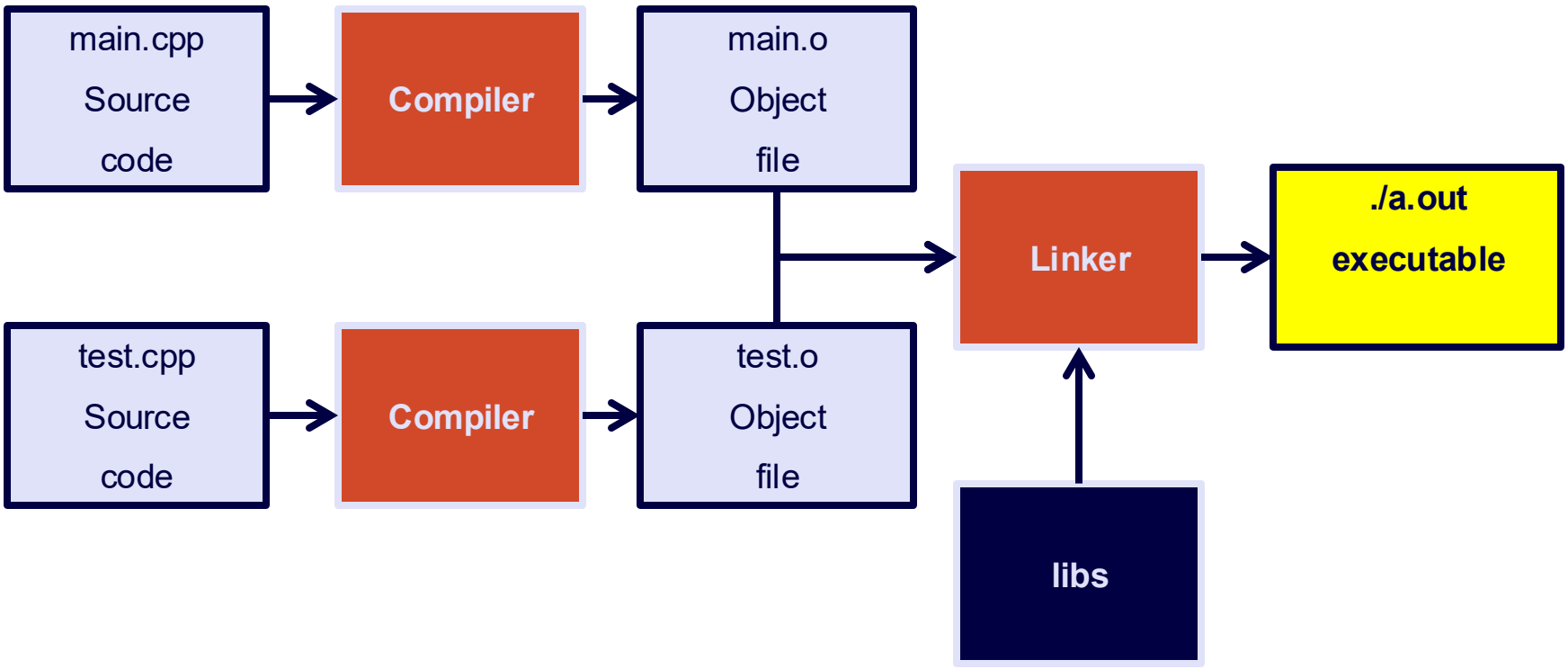
Build Command

- Currently

```
g++ main.cpp
```

- But doing several things behind the scenes

g++ Compile & Link Process



Individual Steps

- First only compile

```
g++ -o main.o -c main.cpp
```

- Then only link

```
g++ -o a.exe main.o
```

- Then run

```
.\a.exe
```

More Friendly Program Names

- First only compile

```
g++ -o main.o -c main.cpp
```

- Then only link

```
g++ -o Lec03.exe main.o
```

- Then run

```
.\Lec03.exe
```

More Complex Programs

- First compile all source code

```
g++ -o main.o -c main.cpp
g++ -o Square.o -c Square.cpp
```

- Then link all object files

```
g++ -o SquareArea.exe main.o Square.o
```

- Then run

```
.\SquareArea.exe
```

Build Process Growing

```
g++ -o main.o -c main.cpp
g++ -o Square.o -c Square.cpp
g++ -o Tri.o -c Tri.cpp
g++ -o Rect.o -c Rect.cpp
g++ -o Circle.o -c Circle.cpp

g++ -o Geometry.exe main.o
Square.o Tri.o Rect.o Circle.o
```

In Practice

- Need to remember which file(s) changed
  - Recompile them (or do all to be safe)
- Relink program
- Manually
- Every time

On Tap For Today

- Building a C++ Program
- Compiler Flags & Directives
- Makefiles
- Practice

Pseuo-Random Process

```
#include <random> // include C++11 random library, why?
using namespace std;

// (1) use random device as seed for generator
random_device rd;
mt19937 mt( rd() ); // effect of seed?

// (2) choose distribution
uniform_int_distribution<int> intDist(iMin, iMax);
uniform_real_distribution<float> floatDist(fMin, fMax);

// (3) generate random number
int randomInt = intDist(mt);         // [iMin, iMax]
float randomFloat = floatDist(mt); // [fMin, fMax)
```

Pseuo-Random Process

```
uniform_int_distribution<int> intDist(14, 19);
```

1. Smallest value generated?
2. Largest value generated?
3. How many different unique values can be generated?

Compiler Flags

```
#include <random> // include C++11 random library
```

- Need to compile against C++11 standard (or newer)

```
g++ -std=c++17 -o main.o -c main.cpp
g++ -o Lec03.exe main.o
```

Makefiles

- Tool to help build a program
  - Uses **`make`*** to automate commands via the terminal
- State what files make up the project
- Will be required to submit with all labs/assignments

- *Note:
  - On Windows, this program is called **`mingw32-make`**.
  - On OS X / Linux, this program is called **`make`**.

make

- Looks in the current directory for a file named **`makefile`** or **`Makefile`**
- Run in terminal via

  `make / mingw32-make`
- Executes the corresponding **`Makefile`**

`Makefile` structure

- Generic format

```
variables = values

target: dependency1
        command1
        command2

dependency1:
        command3
```

- Important note! *command* lines are indented with a tab. Must be a tab.

Dependencies

- **`make`** only executes dependency if timestamps have changed

```
${TARGET}: main.o
        g++ -o $@ $^

main.o: main.cpp
        g++ -o $@ -c $<
```

Cross-Platform `Makefile`

- Demo `Makefile` w/ random numbers

```
make
make clean
make submission
```

On Tap For Today

- Building a C++ Program
- Compiler Flags & Directives
- Makefiles
- Practice

To Do for Next Time

- L1A due before Friday's class
  - Create a Makefile for L1A
- Complete pre-class videos/readings on decision control structures
- **In Class - Random Number Quiz**
  - Closed notes
  - Quick - 4 minutes, 5 questions, MC & FitB
  - Random generation process
  - `cstdlb` **vs** `random`

Covers today's pre-class readings