MAT 685: C++ for Mathematicians

C++: Basics

John Perry

University of Southern Mississippi

Spring 2017

Outline

1. What is C++?
2. Hello C++
3. Compiling
4. Summary

Outline

1. What is C++?
2. Hello C++
3. Compiling
4. Summary

Kinds of computer languages

- Interpreted
  - BASIC, Python, Perl
  - computer reads source, repeats following:
    - translate symbols until full command formed
    - execute command
  - no translation saved

Kinds of computer languages

- Interpreted
    - BASIC, Python, Perl
    - computer reads source, repeats following:
        - translate symbols until full command formed
        - execute command
    - no translation saved
- Compiled
    - C/C++, FORTRAN, Go
    - reads source, translates and saves **machine** code
    - translation works on same architecture (OS, CPU, ...)

Kinds of computer languages

- Interpreted
  - BASIC, Python, Perl
  - computer reads source, repeats following:
    - translate symbols until full command formed
    - execute command
  - no translation saved
- Compiled
  - C/C++, FORTRAN, Go
  - reads source, translates and saves **machine** code
  - translation works on same architecture (OS, CPU, ...)
- Mixed ("bytecode")
  - C#(.NET), Java
  - reads source, translate into **byte**code, saves
  - translation works in "virtual machine" (JVM, .Net, ...)

What sort of work will we do?

C++ is

- compiled
- multi-paradigm
    - focus on object-oriented
- widespread
- difficult, *but*
- rewarding

Which C++

- C++ has several standards:
  - C++98 *(book)*
  - C++03
  - **C++11** *(class)*
  - C++14

Which C++

- C++ has several standards:
  - C++98 *(book)*
  - C++03
  - **C++11** *(class)*
  - C++14

> *Surprisingly, C++11 feels like a new language: The pieces just fit together better than they used to and I find a higher-level style of programming more natural than before and as efficient as ever.*
>
> *— Bjarne Strousroup, inventor of C++*

Which C++

- C++ has several standards:
  - C++98 *(book)*
  - C++03
  - **C++11** *(class)*
  - C++14

> *Surprisingly, C++11 feels like a new language: The pieces just fit together better than they used to and I find a higher-level style of programming more natural than before and as efficient as ever.*
>
> *— Bjarne Strousroup, inventor of C++*

> *C++ does not feel pain. It can't be reasoned with. Starting a fight is a big mistake. If you want to use C++, you must learn to love it the way it is... — C++ FQA*

Which C++ compiler?

Available, installed:

- GCC: GNU Compiler Compilation
  - Gnu's Not Unix
  - industry standard, multi-platform
  - many languages: Ada, C, C++, Fortran, Go, Java, Objective-C, Objective-C++

Which C++ compiler?

Available, installed:

- GCC: GNU Compiler Compilation
  - Gnu's Not Unix
  - industry standard, multi-platform
  - many languages: Ada, C, C++, Fortran, Go, Java, Objective-C, Objective-C++
- LLVM/Clang
  - Low Level Virtual Machine
  - multi-platform, closely tied to Apple, weak on Windows (2017)
  - many languages: C, C++, Fortran, Objective-C
  - extension to use GCC's languages ("dragonegg")

Which C++ compiler?

Available, installed:

- GCC: GNU Compiler Compilation
  - Gnu's Not Unix
  - industry standard, multi-platform
  - many languages: Ada, C, C++, Fortran, Go, Java, Objective-C, Objective-C++
- LLVM/Clang
  - Low Level Virtual Machine
  - multi-platform, closely tied to Apple, weak on Windows (2017)
  - many languages: C, C++, Fortran, Objective-C
  - extension to use GCC's languages ("dragonegg")
- Visual Studio
  - Microsoft, claims to be multi-platform
  - certain languages: C, C++, F#, Python
  - IDE: Integrated Development Environment

Which C++ compiler?

Available, installed:

- **GCC: GNU Compiler Compilation**
  - **Gnu's Not Unix**
  - **industry standard, multi-platform**
  - **many languages: Ada, C, C++, Fortran, Go, Java, Objective-C, Objective-C++**
- LLVM/Clang
  - Low Level Virtual Machine
  - multi-platform, closely tied to Apple, weak on Windows (2017)
  - many languages: C, C++, Fortran, Objective-C
  - extension to use GCC's languages ("dragonegg")
- Visual Studio
  - Microsoft, claims to be multi-platform
  - certain languages: C, C++, F#, Python
  - IDE: Integrated Development Environment

Outline

1. What is C++?
2. Hello C++
3. Compiling
4. Summary

Our version of “Hello, world”

Listing 1: poem.cpp

```cpp
#include <iostream>
using std::cout; using std::endl;

/**
 * A simple program for demonstrating the basics
 * of a C++ project.
 *
 * It does a good job of demonstrating C++
 * fundamentals, but a terrible job with the poetry.
 */

int main() {
  cout << "Don't you just feel like a louse" << endl;
  cout << "To learn that your \"new\" theorem";
  cout << " was proved by Gauss?" << endl;

  return 0;
}
```

Things to notice: setup

`.cpp` book uses `.cc`; can use `.cc`, `.cpp`, `.cxx`

`#include` asks C++ to read another file

`using std::`... tells C++ that any instance of ... comes from a namespace called `std`

- book uses different, discouraged approach
- typically add to beginning of each program

`/*`...`*/` comments, have no meaning to C++

- use `/*` ... `*/` for multi-line comments
- use `//` ... for a one-line comment

...`;` statement terminator (at “statement” end)

Things to notice: general structure

int main() declares a function `main()`

- takes no arguments (nothing in parentheses)
- returns an `int`eger

return 0 return the value 0 to calling program

- "0" convention for "no error"

Things to notice: communication

iostream “library” file for input and output on “streaming” devices

- can only go forwards, not backwards
- printing to a console, to a file, to a printer, ...

cout console **out**put

<< operation on `cout` and following text string

” quote to begin and end a text string

\” “escape” character to include quotation mark in text string

endl **end**-of-**l**ine object

Outline

1. What is C++?
2. Hello C++
3. Compiling
4. Summary

Invoke g++

Invoke the compiler

```
$ g++ <options> <filename>
```

Invoke g++

Invoke the compiler

```
$ g++ <options> <filename>
```

Execute your program

- `<options>` include an “object name”

  ```
  $ <object name>
  ```

- `<options>` do not include an ”object name” (naughty!),

  ```
  $ a
  ```

  (seriously)

Compiler options

-o	name of executable, follow with `<object name>`
-g	include debug information
-pg	include profiling information
-Ofast	optimize for highest speed
-std=c++11	use the C++11 standard

Example

To create a debuggable executable `my_poem` from `poem.cpp`:

```
$ g++ -o my_poem -g poem.cpp
```

To execute:

```
$ my_poem
```

To debug it: (we talk about this later)

```
$ gdb my_poem
```

Basic debugger commands

`b` *i* create a breakpoint at line *i* (this allows you to stop the program at a line of interest)

`n` execute the next line in the program

`run` let's get this party started

`until` *i* execute until we reach line *i*

`quit` I hate to be a downer, but...

Outline

1. What is C++?
2. Hello C++
3. Compiling
4. Summary

Summary

- C++ is a compiled, multi-paradigm, object-oriented programming language
- Using C++11
- basic C++ features illustrated in `poem.cxx`

Homework

pp. 8–9 #1.1–1.5