

# Assessment – C++

Jim Fawcett

CSE687 – Object Oriented Design

Summer 2017

# C++ Strengths

- ***Is very powerful***
  - Can program at very low level or very high level of abstraction
  - Has direct access to memory
  - User defined objects are first class citizens
  - Supports structured design, object oriented design, and generic design
  - Has a flexible memory model – static, stack, and heap
- ***Language emphasizes performance***
  - You don't pay for features you don't use
  - Almost no runtime checking
- ***Is platform agnostic***
  - Language and standard library apply to all platforms
  - Easy to isolate platform dependencies in small modules.
- ***Has a governing standard***
  - C++11, C++14, C++17

# C++ Strengths

- **Has many useful features**

- Callable objects – function pointers, functors, and lambdas
  - Defining thread operations
  - Defining operations to apply to STL containers
- Templates
  - Support specialization
  - Enables defining functions that accept a wide variety of types
- Deterministic resource management
  - Allocate resources in constructors
  - Return them in destructor
  - All that happens automatically, not requiring any special handling by using code

- **Has very rich library ecosystem**

- Strings, STL, Streams
- Threads, atomics
- Regular expressions

# C++ Strengths

- **Is very expressive**

- Namespaces and type aliases allow you to endow your code with semantic meaning
- Operators help you code algorithms with syntax close to what you use to describe them on paper
- Binding lambdas to `std::function` allows you to provide locally named blocks of processing that you can use in several places in a function or class methods.

# C++ Strengths

- **C++ is widely used**

- System Programming
- Applications
- Scientific Programming

- **There are a lot of resources available:**

- [Stroustrup's Site](#)
- [Herb Sutter's Site](#)
- [CodeProject](#)
- [Boost C++ Library](#)
- [C++ at Microsoft](#)

# C++ - Weaknesses!

- ***C++ is a complex language***

- Context dependent

- For example, the keyword `static` has four distinctly different meanings, depending on the context of its use.

- Has some rules not enforced by the compiler

- Use virtual destructors
- Use initializers in constructors
- Don't return references to temporaries

- Some rules have exceptions

- Compiler will automatically generate constructors:
  - All constructors? No. Which ones? That depends.
- In fact, the language implies many implicit operations.
  - The good news is that C++ does lots of things for you!
  - The bad news is that C++ does lots of things for you!

# C++ Weaknesses?

- ***Safety is not automatic***

- No bounds checking
- Direct access to memory
- Can break the type system with casts
- Thrown exceptions can leave program in undefined state.

- ***However, you are in control!***

- You want bounds checking – then build it into your class – it's not hard.
- You can create smart, safe pointers if you want – the C++11 standard library has done that.
- You can safely cast and write exception safe code. You just have to know how!
  - That's why, among other reasons, that you are here.

# C++ - Omissions

- ***Things not supported by the language***

- Directory services
- Graphical User Interfaces
- Network programming
- Internet programming
- Database operations

- ***You can provide all those things.***

- You will find FileSystem and Sockets packages on the college server
- You'll see how to make C# WPF code interoperate with native C++



# Resources To Help You

- **Class Texts**
- **Website:** <https://ecs.syr.edu/faculty/fawcett>
- **For OnLine Students:**
  - Weekly Synchronous lectures
- **For residential students:**
  - **Friday help sessions:**
    - 9:00am – 10:30am in CST 4-201 by instructor
    - 5:30pm – 7:00pm in CST 4-201 by TAs
  - **Teaching Assistants:**
    - Office Hours
  - **Special Help Sessions**
    - One or two sessions covering Visual Studio IDE

**End of Assessment**