C03: Quickstart Guide for Class C++ Programming Assignments

ETAMU Class Assignments

Derek Harter

Department of Computer Science
East Texas A&M University

Summer 2025

Outline

- Compiling, multi-file projects
- Standard I/O streams
- Using c++ strings
- Unit testing in C++ with catch2 framework
- OO Programming w/ C++ classes
- STL
    - Vector and List Containers
    - Maps
    - Queues, Stacks, Priority Queues
    - Iterators and Algorithms

Compiling, Command Line, Multi-File Projects

- We are using `make` build tool

```
$ make clean
$ make all
$ make unit-test
```

- Multi-file C/C++ projects should
  - Put implementations into source file '.cpp'
  - Put declarations and prototypes into header file `.hpp`
  - Use header guards or `#pragma`
  - Use `#include` preprocessor directive to reuse code
- C/C++ has a single `main()` function that serves as entry point to executable
- Can pass command line arguments
  - `argc` the argument count
  - `argv` the argument values.

Using the C++ `string` type

- You should never use old C character arrays to represent strings in C++ unless you are absolutely forced to for some reason.
- `string` type provides much higher level abstraction of string processing and representation.
- You can concatenate strings, but use string streams if need to format a string.

Using the C++ Streaming Library for I/O

- `#include <iostream>` for basic streams including standard input `cin` and standard output `cout`
    - `<fstream>` provides file input and output streams.
    - `<sstream>` provides string streams
- Can format streams using `<iomanip>` stream manipulators
- Also check that a file opened correctly before using a file stream.
- Use string streams instead of string concatenation when you need more advanced string formatting capabilities.

Object-Oriented programming with C++ Classes

- Classes in OOP encapsulate the state of a single object instance
- Typically all member variables are `private` to protect that state, only member functions of the class can see or modify the state.
- The class declaration should go in the `.hpp` file, and all class member functions in the corresponding `.cpp` file.
- Provide “setter” and “getter” methods and member methods to expose and manipulate an instances state.
- The basic form of memory management using C++ classes is that memory is allocated (`new`) in the class constructor, and it is freed up (`delete`) by the class destructor.

Using C++ Standard Template Library

The C++ Standard Template Library STL provides high-level data types and algorithms, bringing modern C++ on par with high level languages like Python and Java.

- Use C++ containers like `list`, `vector` and `map`
- C++ iterators are abstraction to "iterate" over elements of a container, they are used extensively so should be familiar with the concept.
- C++ STL provides Algorithms and Functors, not just container data types.

C++ Standard Template Library: Sequential Containers

- `array` provides a container that is a wrapper around an old C-style array, but does have benefits (like `size()` member function).
- `vector` can grow dynamically when you add items to front or back. Does use a block of memory, so can have bad performance when needs to grow memory allocation.
- `list` also can grow dynamically and efficiently when adding to front or back. implemented basically as a doubly linked list

C++ Standard Template Library: Associative Containers

- If you are familiar with Python or Java, you may have used Python Dictionary or Java `HashMap`
- C++ STL `map` provides an associative key-value pair container.
  - Can insert and retrieve by keys, like `string` or other types.
  - A `map` is actually an ordered associative container, it uses somtehing like a binary tree so that you can iterate over it in sorted key order.
- C++ STL `unordered_map` is also an associative key-value pair container
  - It uses a hash table, so keys are not stored in any particular order and you cannot efficiently iterate over an unordered map in sorted key order.

Additional Recommended Resources

The following are suggested online materials you may use to get more information and help about learning the basics and more advanced aspects of programming in C++:

- cplusplus.com Tutorials and Reference Has both some okish tutorials on C++ and a pretty good reference for C and C++ librarys, including STL.
- Programiz.com Introduction to C++ has tutorials / courses you can pay for, but the free overviews are helpful quick starts for people who know how to program to pick up modern C++ and the STL

Bibliography