ECE 2574: Data Structures and Algorithms - Class Design and Testing

C. L. Wyatt

Today we will continue working on our Bag template using a static array internally and define more tests using CMake and Catch.

- Review the Bag API
- Warmup
- Write more Bag tests
- Implementing the Bag ADT using a static array

Warmup #1

What is the difference between a struct and a class in C++?

- There is no difference (1%)
- Default visibility is public for classes and private for structs (7%)
- Default visibility is private for classes and public for structs (91%) CORRECT
- struct are only for C not C++ (0%)

Warmup #2

Why would I pass a parameter by constant reference rather than by value?

- To make things complicated (0%)
- So that the parameter can be changed by the function (32%)
- It is always faster (26%)
- To save a copy being invoked (42%) CORRECT

Warmup #3

In C++ can I have a class member variable named x and a class method/member function named x()?

- True (48%)
- False (52%) CORRECT

Warmup #3

In C++ can I have a class member variable named x and a class method/member function named x()?
The reason you can't is because in C++ you can take the address of both variables and functions (both are just memory locations after all).
Example:

```
struct Foo{
  int x;
  int x(){ return x; };
};

Foo myfoo;
call_a_func(&myfoo.x);
```

On the last line, the compiler cannot figure out if you are taking the address of the **member variable** x, or the **member function** x.

Exercise: Lets write some more Bag tests and get them to compile and run.

1. Download starter code from website
2. Open `bag_tests.cpp` and add some tests for each method of bag
3. Build and run the tests (they should fail)

Building code and running tests using CMake and various IDEs

CMake is a tool that generates the project files for several IDEs from a configuration file named CMakelists.txt.
So the workflow is:

1. You run CMake, giving it the location of the **source directory** and where you want the build artifacts to go, the **build directory**.
2. You hit configure and select which IDE or build tool you wish to use, then, if everything is ok, generate.
3. This will put all the necessary files into the build directory.

Just double click on the .sln file to open it is VS. If you are using XCode double click on the .xcodeproj file. If you want to use the command line you can invoke cmake with the build option, `cmake --build BUILDDIR`, this will invoke the command-line build tool configured.
See demo or workflow tutorial on the website.

Exercise: starting an implementation using automatic array storage

4. Implement the Bag using a fixed size array internally in `bag.hpp` and `bag.tpp`
5. Upload your files `bag_test.cpp`, `bag.hpp` and `bag.tpp` to Canvas under “Exercise for Meeting 3”

Next Actions and Reminders

- No class Monday due to Labor Day Holiday
- I will have only intermittent online access this weekend
- Read CH pp. 37-46 on basic polymorphism
- Complete the warmup before noon Wednesday 9/6
- Program 0 is due Friday 9/8 before 11:55 pm