Final Review

Basics

The final will be Monday, May 5th at 10:30 am. It will, unlike the midterm, be in the RGAN building room 211.

You will have 50 minutes for the exam, just to give you a little bit of leeway when it comes to getting to RGAN 211, getting started, and getting the second shift started. The exam is designed with this time limitation in mind. The time is strictly monitored, though – the system will lock you out 50 minutes after *you* start your exam. This is done to ensure fairness; I can't keep the lab past 12:30 and need to split the exam into two "shifts".

If your last name starts with A-H, your exam will be from 10:30-11:30; if your last name starts with J-Z, your exam will be from 11:30-12:30. If your last name starts with I, you're not on my roster.

Anticipate no more than 40 multiple choice questions and no more than 5 short answer questions, and no more than 40 total questions.

Comprehensiveness

The final is comprehensive. There will, however, be more focus on the second half of the semester. Given that most of the second half of the class was more focused on implementation of PA3, the final itself is not going to be terribly long.

What you need to know

For starters, it's a good idea to read (or, really, skim) everything that *I* have written – the practicum assignments, homeworks, etc.

Notably, understand the choice between implementing Entity position in PA3.

Specific topics

- Everything is an expression is still very important
- Everything on the midterm is in play
- Typing is still quite important
  - Especially understand that C++ is statically typed, which means the type of any expression is known *exactly* at compile time.
    - Note! The type of an expression can be a pointer to a base class, but the pointer can point to an object of a derived class – the difference here is important to understand!
    - (i.e., an *expression*'s type may not be exactly the same as the type of the expression's *value* at run time -- although unless you've done something really wrong they will be compatible)
  - Make sure you're aware that pointers and references are part of type -- e.g., `int` is different from `int*` which is different from `int&`, `int**`, and `int*&`.

    - You should know what `int`, `int*`, `int&`, `int**`, and `int*&` all mean
    - `int***` too
    - And `int*****` while we're at it.
- Design patterns, particularly:
    - Singleton
    - Class factory
    - Iterator
    - And, really, what a design pattern *is*
- Exception handling (albeit not in great detail; just know what it is and be able to understand the technique if it is described)
- Templates/generic programming
    - You don't need to exacting details of syntax
    - But be able to read template code and have an understanding of where a template could be useful
- Abstract classes
    - Pure virtual methods
    - Interface
- Explicit casting
- Forward declaration
    - How to do it/what it looks like
    - What kind of problems we solve with it
- Include guards
- Operator overloading
    - Again, no exacting detail of syntax, but understand what it does
- Libraries
    - Know the difference between static and dynamic linking

What you don't need to know

Any history covered in class; for the final, you don't need to know any of it. I'm not even going to ask another question about *Go To Considered Harmful*.

You will not be asked to write any code, but you will need to be able to read code. You *will* (still) need to know a lot of stuff from CS215 to be able to read the code presented -- `<iostream>` objects, vectors, etc. Make sure you can make sense of the code examples posted on the web site.