TEST 2 STUDY GUIDE

MAT 685, C++ FOR MATHEMATICIANS

The test will cover **from Chapter 5 through Chapter 10**. *However,* I will ask nothing that is specific to the book from Chapter 7, and I gave a radically different approach to the same material as Chapter 10. So don't worry about those two chapters, and in particular for Chapter 10 review instead the last set of slides.

Keep in mind that it is generally more important to understand the *ideas* than the *details*. You can always look up details in a book or online. Ideas are not so straightforward.

COMPUTER SCIENCE

Data structures.

- native arrays
  - what an array "looks" like
  - static v. dynamic creation
- classes
  - basic goals of object-oriented programming: what do these mean? how does C++ address them?
    - data hiding
    - encapsulation
    - inheritance
    - overloading
    - polymorphism
  - fields
    - access specifiers (`public`, `protected`, `private`)
    - static fields of a class
      - where to initialize them
      - difference with static variables in a function
  - constructors
    - default, copy constructors: what are they? what should they do?
    - initializer list
      - the idea
      - client's point of view
      - constructor's point of view
    - `new []` and `delete []`
    - "construct before you compute"
  - methods
    - what are they?
    - difference between a *method* and a *function*
    - `const` methods

Date: Spring 2017.

        - defining methods in a class declaration v. outside
            - necessity of `Class_name::method_name(...)` in latter case
        - operators
            - difference between pre- and post-inc/decrement
- containers
    - where to find them / how to use them
    - `vector` v. `array` v. native array v. `list`
        - how do we declare them?
        - how do we put stuff in them?
        - advantages/disadvantages of each?
    - don't worry about `set`, `unordered_set`, `map`, `unordered_map`
    - iteration
        - iterating without iterators
        - iterating *with* iterators
            - obtaining, using
            - `iter` v. `const_iter`

Basic structure of a C++ program.

- namespaces
    - what's the point?
    - how defined?
- `auto` type
- exceptions: how to throw, catch
- templates
    - templated function
    - templated class
    - information must be known *at time of compilation*
- how do I include/link an externally-compiled library?
- which language features are new to C++11

MATHEMATICS

Euler's totient function.

- basic formulas
- relationship to factorization

Sieve of Eratosthenes.

- method
- when to quit
- utility for Euler's totient function

Chinese Remainder Theorem.

- idea
- how to solve it

Modular arithmetic.

- relationship to gcd/Bézout's identity (i.e., invertibility, if that's a word)
- fast exponentiation by recursively dividing the exponent

Hierarchy of rings.

- commutative v. non-commutative
- modulo m, modulo p

Matrices and polynomials.

- defined over a ring
- difference between dense, sparse representations (the idea, not the details)

Online Encyclopedia of Integer Sequences.