COMP 2404 Midterm Review – Fall 2018

- Section 1 -- Basics of C++ development
  - Linux platform
    - Types of shells
    - Basic Unix commands, I/O redirection
    - Types of Unix platforms
    - Program building
      - Compiling vs linking: what are they, what commands are used
      - Source files, object files, executables
      - Makefile: what are they, what are they used for
  - Basic language features
    - Terminology:
      - Expressions, statements, blocks, scope (global, local)
      - Operators, operands, arity, precedence, associativity
    - Functions
      - Global vs member functions
      - Function declaration vs implementation
      - Function design
    - Types of parameters (input, output, input-output)
    - Parameter passing (pass by value, pass by reference by reference, pass by reference by pointer)
    - References: what are they, what are they not, how are they used
  - Programming conventions
    - Naming conventions, indentation, commenting

- Section 2 -- Basics of C++ classes
  - Class definition
    - Binary scope resolution operator ::
    - Access specifiers (public, private, protected)
    - Code organization: header file, source file
    - Class interface: how you interact with a class, set of public members
  - Constructors and destructors
    - Default arguments
    - Types of constructors: copy ctor, conversion ctor, default ctor
    - Destructors: order of execution
    - Copy constructors: what are they, what they do, when they are called

- Memory management
  - Stack vs heap
  - Pointers
    - What are they, why are they used, how are they used
    - Operators: arrow (->), dereferencing (*), address-of (&)
    - Differences between pointers, references
    - Parameter passing with pointers
    - Memory allocation: static vs dynamic
    - Memory leaks
    - Dynamic memory allocation: new, delete
    - Arrays (4 kinds of arrays):
      - Statically allocated array of object pointers
      - Statically allocated array of objects
      - Dynamically allocated array of objects
      - Dynamically allocated array of object pointers

- Section 3 -- Basics of object-oriented design
  - Software engineering overview
    - Software development life cycle: requirements analysis, design, implementation, testing
    - OO design principles
  - Information hiding
    - Data abstraction: separating class interface from implementation
    - Encapsulation: grouping together data that belongs together
    - Principle of least privilege
  - Object design categories
    - Types of objects: control, view, entity, collection
    - What are they, what do they do, what are they responsible for, why? Maintainability
    - Collection classes
  - Documenting design
    - UML class diagrams
      - Not collection classes, no getters, setters, ctor, dtor
      - Classes: attributes, operations
      - Associations: composition, inheritance
      - Composition: directionality, multiplicity

- Section 4 -- Essential object-oriented techniques
    - Encapsulation
        - Composition:
            - Member initializer syntax
            - Order of ctor, dtor
        - Constants (objects, data members, member functions)
        - Friendship
        - Static class members
        - Linked lists (insertion, deletion, cleanup)
            - Singly, doubly, with/without a tail
    - Inheritance
        - Terminology: base vs derived
        - Member access
        - Base class initializer syntax
        - Ctor, dtor, order in which they execute

Midterm!

- Up to section 4.2 (inheritance) slide #11
- 80 minutes
- Out of 50 marks
- Multiple choice (4 questions): 8 marks
- Concept exercise: 20 marks
- Programming (2 questions): 22 marks

BRING:

- Campus card
- Pencils, erasers

No questions ☹