COMP 2404 Final Exam Review – Fall 2018

- Section 1 -- Basics of C++ development

  - Linux platform
    - Types of shell
    - 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
      - Makefiles

  - Basic language features
    - Terminology
      - Expressions, statements, blocks, scope (local vs global)
      - 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 they are, what they are not, how they are 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: separating code into header and source files
    - Class interface: how you interact with a class, the set of its public members

  - Constructors and destructors
    - Default arguments
    - Types of constructors: default ctor, copy ctor, conversion ctor
    - Destructors: order of execution
    - Copy constructors: what are they, what they do, when are they called

    - Memory management
        - Stack vs heap
        - Pointers
            - What are they, why are they used, how are they used
            - Operators: arrow operator, address-of operator, dereferencing operator
            - Differences between pointers and references
            - Parameter passing with pointers
            - Memory allocation: static vs dynamic
            - Memory leaks
            - Dynamic memory allocation: new, delete
            - 4 kinds of arrays:
                - Statically allocated array of objects
                - Statically allocated array of object pointers
                - 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 (development process): requirements analysis, design, implementation, testing
        - OO design principles

    - Information hiding
        - Data abstraction: separating implementation from class interface
        - Encapsulation: grouping data together that belongs together
        - Principle of least privilege

    - Object design categories
        - Types of objects: collection classes, control, view, entity
        - What are they, what do they do, why?
        - Collection classes

    - Documenting design
        - UML, UML, UML
            - Classes: attributes, operations (access specifier)
            - Associations: composition, inheritance
            - Composition: directionality, multiplicity
            - Inheritance: abstract or not
            - NO COLLECTION CLASSES, NO GLOBAL FUNCTIONS
            - No getters, setters, ctors, dtors

- 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 class, derived class
        - Member access
        - Base class initializer syntax
        - Ctor, dtor order of execution
        - Types of inheritance: public, private, protected
        - Multiple inheritance: diamond problem (multiple inclusion, virtual inheritance)

    - Design patterns
        - Types: structural, behavioural, creational
        - Façade, Factory, Observer, Strategy, anti-patterns

    - Polymorphism
        - What is polymorphism
        - Dynamic binding
        - Virtual functions
        - Abstract classes using pure virtual functions
        - Behaviour classes, Strategy design pattern

    - Overloading
        - Function overloading
        - Operator overloading
        - cascading

    - Templates
        - Function templates
        - Class templates (collection classes)

    - Exception handling
        - Dealing with faults, fault prevention, fault detection, fault tolerance
        - Why exception handling vs inline error handling
        - Try, throw, catch
        - Stack unwinding

- Section 5 -- C++ library

  - STL
    - Iterators
    - Sequence containers (vector, list, deque)
    - Associative containers (map, set)
    - Container adapters (stack, queue)
    - algorithms

  - Files and streams
    - Input streams
    - Output streams
    - Files
    - Error state flags

- Final exam
  - 3 hours
  - Covers all the course material
  - Out of 100 marks
  - Concept questions (all multiple choice):   40 marks
    - 40 questions, 1 mark each
  - Programming and UML: 60 marks
    - 4 questions

  - BRING:
    - Campus card
    - Pencil, erasers

  - ASSIGNED SEATING !!!!!!   Check Grades in cuLearn on Monday for *Row* and *Seat* numbers

  - NO QUESTIONS -- sorry! :-(