# COMP 2404 Final Exam Review – Winter 2019

- Section 1 -- Basics of C++ development

  - Linux platform
    - Types of shells
    - Basic shell commands
    - Program building:
      - Makefile: what are they, what are they used for, commands
      - Compiling, linking: what are they, what commands are used
      - Source files, object files, executables

  - Basic language features
    - Variables
    - Functions:
      - Global vs member functions
      - Function declaration vs implementation
      - Function design
    - Types of parameters, parameter roles
      - Input, output, input-output parameters
    - Parameter passing: pass by value, pass by reference by pointer, pass by reference by reference
    - Operators: operands, arity, precedence, associativity
    - Expressions, statements, blocks, scope (local vs global)
    - References: what they are, what they are not

  - Programming conventions
    - Naming conventions: constants, variables, functions, data types
    - Indentation
    - commenting

- Section 2 -- Basics of C++ classes

  - Class definition
    - Binary scope resolution operator
    - Access specifiers (public, protected, private)
    - Code organization: header files vs source files
    - Class interface: set of public members of a class
    - Include guards

  - Constructor and destructors
    - Default arguments
    - Default constructor: what are they, order of invocation
    - Destructors: what are they, order of invocation
    - Copy constructors: what are they, when are they called

- Memory management
  - Stack vs heap
  - Pointers:
    - What are they, why are they used, how are they used
    - Pointer operations:  address-of (&), dereferencing (*), arrow (->)
      - stuPtr->age == (*stuPtr).age
    - Differences with references
    - Parameter passing with pointers
    - Memory allocation:  static vs dynamic
    - Memory leak
    - Dynamic memory allocation:  new, delete
    - Arrays:
      - Dynamically allocated array vs statically allocated
      - Arrays of objects vs arrays of object pointers
      - How to allocate and deallocate all 4 kinds of arrays

- Section 3 -- Basics of object-oriented design

  - Software engineering overview
    - Life cycle
    - OO design principles

  - Information hiding
    - Data abstraction:  making interfaces simple, separating interfaces from implementation
    - Encapsulation:  grouping together data and behaviour that belongs together
    - Principle of least privilege

  - Object design categories:
    - NOT MVC (model-view-controller)
    - Types of object categories:  entity, control, boundary (view, UI), collection classes
    - What are they, why do we separate them

  - Documenting design
    - **UML class diagrams:**
      - Classes:  attributes, operations (parameters and type – in/out/in-out)
      - association, relationships:  inheritance, composition
      - composition:  directionality, multiplicity
      - don't show: collection classes, getters/setters (simple), ctor, dtor, other objects as attributes, friendship

- Section 4 -- Essential object-oriented techniques
    - Encapsulation
        - Composition: member initializer syntax, ctor & dtor order
        - Constants (objects, data members, member functions)
        - Friendship (almost NEVER use it)
        - Static class members
        - **Linked lists**: singly linked, doubly linked, with/without 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, 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, make sure cleanup is done

- Section 5 -- C++ library
  - STL
    - Iterators
    - Sequence containers (vector, list, deque)
    - Associative containers (map, set)
    - Container adapters (stack, queue, pqueue)
    - algorithms
  - Files and streams
    - Input streams
    - Output streams
    - Files
    - Error state flags

- Final exam
  - Details:
    - 3 hours
    - out of 100 marks
  - Concepts: 40 marks
    - MCQ: 34 questions, 1 mark each
    - Exercise: 1 question, 6 marks
  - Programming: 60 marks
    - 5 questions
  - BRING:
    - Campus card,
    - pencils, erasers
  - ASSIGNED SEATING: under Grades, by April 22, “Row” and “Seat”
  - NO QUESTIONS :-(