COMP 2404 Review – Winter 2014

- Section 1 -- Basics of object design
  - Software engineering activities
    - Requirements analysis (kind of)
    - Design
    - implementation
    - testing
    - UML notation
  - C++
    - Classes
    - References&pointers
    - Memory management & dynamic memory allocation
    - Constructors (default, copy, conversion)
    - Destructors
    - Separate code into source and header files

- Section 2 -- Data abstraction
  - Encapsulation
    - Access modifiers
    - Principle of least privilege
    - Friendship
    - Composition
    - Static class members
  - Inheritance
    - Base class vs derived class
    - Types of inheritance (public, protected, private)
    - Multiple inheritance (virtual base class vs. multiple inclusion)
  - Polymorphism
    - Abstract classes
    - Virtual, pure virtual functions
    - Dynamic binding
  - Overloading
    - Function overloading
    - Operator overloading
    - Cascading

- Design patterns
  - Façade, Observer, Factory, Strategy
  - Established techniques to deal with specific situations or problems
  - Allow for better reusability, extensibility/modifiability, scalability
- Interface classes
  - Encapsulating behaviour, Strategy design pattern
  - Encapsulating object creation, Factory design pattern

- Section 3 -- Code reuse
  - Templates
    - Function templates
    - Class templates
  - STL
    - Containers
      - Sequence containers (vector, list, deque)
      - Associative containers (set, map, etc.)
      - Container adapters (stack, queue, priority queue)
    - Iterators
    - Algorithms
  - Files and streams
    - File I/O
    - Error bits (good, bad, fail, eof)

- Section 4 -- Software robustness
  - Dealing with faults
    - Fault prevention
    - Fault detection
    - Fault tolerance
  - Exception handling
    - Try, throw, catch
    - Unwind the stack (cleaning up resources)

STUDENT GUESSES ABOUT EXAM QUESTIONS:

- I give you code, you fill in the functions or parts of the code
- Coding question with polymorphism with Timmy and Harold
- Differences between concepts
- Describe a concept and tell me why it's important in software engineering
- You make a UML diagram (with the works)
- Describe a concept and tell me if it's good software engineering or not
- I give you a UML diagram, and you write code or class definition
- You give me a prototype for overloaded operator
- Bonus question ☹
- Draw pictures of farm animals – NO!!!
- Tell me about copy constructor vs other types of constructors
- Abstract classes
- Virtual vs pure virtual functions, what makes a class abstract?
- Which overloaded operators re-use each other?
- Coding question: take into account memory management (no leaks)
- Given code, is it using the copy constructor or assignment operator?
- Code a container class
- Describe two or three types of design patterns
- Virtual destructors and memory management?
- Give you code or UML diagram, and ask you which classes could be made abstract
- Files and streams
- Or not!

EXAM:

- Out of 100%
- Part A – Concepts (30%)
- Part B – Programming (70%)
    - Concept-like questions (22%)
    - Writing code (48%)
- Exclude: STL associative containers, STL container adapters