

# Semester Review

---

7 Dec 2010

CMPT140

Dr. Sean Ho

Trinity Western University

# Semester review: outline

- Problem Solving: Software Development
- Python basics
- Functions
- OO design
- Additional language tools

# CMPT140: Problem Solving

- Roles and Relationships:
  - Client ↔ Proj. Mgr ↔ Architect ↔ Coders
  - Requirements → Specification → Design → Implementation → Testing → Deployment
- Development Models
  - Waterfall (WADES), V
  - Agile: Spiral, Scrum
- Design Concepts: data model, class diagram, functional spec., pseudocode, stub design
- Testing Concepts: unit testing, integration testing
  - Test-driven development, pre-/post-conditions

# Python basics

- Interpreter, compiler, virtual machine
- Variables, built-in types, aliasing
- Console I/O: `input()`, `print()`
- Operators: arithmetic, Boolean, type conversion
- 5 control/flow abstractions
  - `if/elif/else`
  - `while/for, continue/break/else`
- Strings, formatting
- `import, from * import *`, math library
- `range()`

# Functions

- Modular design, top-down problem solving
- Defining functions, invoking functions
  - Formal parameters, actual parameters
  - Call-by-value vs. call-by-reference
- Call stack, stack frame
- Recursion
  - Factorial, Fibonacci

# OO Design

- **Terminology**: class, instance, method, attribute
- **Using** classes: calling **constructor**, using **methods**, assignment (aliases)
- **Designing** classes: **attributes**, **methods**
- **Creating** classes: **constructor**, **\_\_str\_\_**, set/get
  - **self**, private attributes, default values

# Additional language tools

- **Lists** (building, indexing, iterating, operations, passing to functions)
- **Dictionaries** (building, indexing, iterating, ops, ...)
- **Exceptions: try/except/finally**
- **File I/O: open(), with, modes, reading/writing text**
  - **Serialization (pickle)**
- **Drawing using graphics.py**
  - **Classes for drawing objects, getMouse(), Entry**