

CMPT14x Semester Overview

3 Dec 2008

CMPT14x

Dr. Sean Ho

Trinity Western University

Quiz09

- Declare a Python **class** defining a node of a **circular doubly-linked list**
 - Draw a **diagram** representing a circular doubly-linked list with **four** nodes
- What does it mean for a binary search tree to be **balanced**? Why is this a **good** thing?
- **Insert** these items into a new BST in the following order:
6, 3, 4, 1, 12, 8, 10, 9
 - Now **delete 6** from the tree (two options)
- **add(1), add(2), get(), add(3), get(), add(4), get(), get()**
 - What do the **get()**s return if this is a
(a) **stack**, (b) **queue** ?

Review: major concepts

- Problem solving / design process
- Program elements:
 - Expressions, sequence/if/while/etc.
- Program organization:
 - Procedures, modules, libraries, scope
- Data types: lists/arrays, classes, sets, dict
- Standard libraries: math, random, file I/O
- Exceptions
- OO concepts

CMPT 140: M2 chs 1-8

- Ch1: Problem-solving
- Ch2: Your first program
- Ch3: Program structure
- Ch4: Procedures/functions
- Ch5: Arrays/lists
- Ch6: Library modules: math, random
- Ch7: Applications
- Ch8: Number bases and memory/storage

CMPT 145: M2/Py

- Sets (M2 ch9)
- Dictionaries (Py ch10)
- Object-oriented programming (Py ch12-14)
- Exceptions (Py tut 8)
- Namespaces and scope (Py tut 9)
- Pointers and linked lists (M2 ch12, Py ch17)
- Trees, BST, Stacks/Queues (M2 ch14, Py ch20)

Where to go from here?

- Now you know the **concepts**; learning C/Java/M2/etc. is mostly just learning **syntax**
- Learn by **example**:
 - Find a small, well-written application and
 - Figure out how it works; read the **code**
- Learn by **doing**:
 - Modify/extend, or **create** your own app!
- Next semester: sign up for **CMPT166**
 - **C++** in Visual Studio and Eclipse