Ch1-7 Review

22 Oct 2007
CMPT14x
Dr. Sean Ho
Trinity Western University

Today: Chapters 1-7 Review

- Ch1: Problem-solving

- Ch2: Your first program

- Ch3: Program structure

- Ch4: Procedures/functions

- Ch5: Arrays/lists

- Ch6: Library modules

- Ch7: Applications

TRINITY WESTERN UNIVERSITY

Ch1: Problem solving

- Computing scientists as toolsmiths
- Top-down vs. bottom-up; WADES
- Client --> Designer --> Implementer
  - Requirements doc, Design spec, Code
- Design, pseudocode, documentation
- Abstract data types
  - Atomic vs. compound
  - What's the difference: 5, 5.0, '5', (5), {5}
- 5 hardware abstractions, 5 control/flow abstractions

TRINITY
WESTERN
UNIVERSITY

Ch2: A basic Python program

- Components of a baby Python program
- Literals, identifiers and reserved words (examples?)
- Strings, quoting, newlines
- Statically-typed vs. dynamically-typed
- Declaring and initializing variables
- Keyboard input: input(), raw_input()
- Expressions, operators, and precedence rules
- Formatted output: %d, %f, %s

Ch3: Basic Program Structure

- Statement sequences

- Selection (if, else, elif)

- Repetition/loops (while, for)
  - Top-of-loop vs. bottom-of-loop testing
  - Sentinel variables
  - continue, break, else

- Sequence concatenation (+) and repetition (*)
  - Works on strings, lists, tuples

TRINITY
WESTERN
UNIVERSITY

Ch4: Functions

- **Procedures** (functions, subroutines)
  - **No** parameters
  - With **parameters**
  - **Formal** vs. **actual** parameters
  - **Scope**
  - **Global** variables (why not to use them)
  - Call-by-**value** vs call-by-**reference**
    - Python is call-by-**object**, which is like call-by-**value** for **immutable** types and call-by-**reference** for **mutable** types

Ch5 (+Py ch8): Arrays and Lists

- Call stack, backtrace

- Python lists vs. M2/C arrays

- Lists as function parameters

- Multidimensional arrays/lists

- Python-specific list operations
  - Membership (in)
  - Concatenate (+), repeat (*)
  - Delete (del), slice ([s:e])
  - Aliasing vs. copying lists

TRINITY
WESTERN
UNIVERSITY

Python type hierarchy (partial)

- **Atomic types**
  - **Numbers**
    - **Integers (int, long, bool): 5, 500000L, True**
    - **Reals (float) (only double-precision): 5.0**
    - **Complex numbers (complex): 5+2j**
- **Container (aggregate) types**
  - **Immutable sequences**
    - **Strings (str): "Hello"**
    - **Tuples (tuple): (2, 5.0, "hi")**
  - **Mutable sequences**
    - **Lists (list):   [2, 5.0, "hi"]**
  - **Mappings**
    - **Dictionaries (dict): {"apple": 5, "orange": 8}**

TRINITY WESTERN UNIVERSITY

Ch6: Standard I/O and Libraries

- Working with files: file objects, open(), close()
  - Input: read(), readline(), readlines()
  - Output: write(), flush()
  - The file position pointer: seek(), tell()
- Standard I/O channels: sys.stdin, stdout, stderr
- Python standard math library

- Libraries: interface (DEF) vs implementation (IMP)
- Accessor (set/get) functions

Ch7: Applications

- Null-terminated strings; lexical sorting

- fractions.py ADT library:
  - Set/get functions to hide tuple implementation

- substitution.py cipher library:
  - How it works, encode/decode

- pseudorandom.py RNG library:
  - Seed, iterative process
    - (Understand concepts enough to code it)
  - Testing via histograms

TODO items

- CMPT140 Final ch1-8 this Wed-Thu 24-25Oct
  - Everyone attend Thu class, even 141/143!
- Register for CMPT145 if you haven't already
- No quiz/homework this week
- Lab06 due next week Wed: ch7 (choose one):
  - # 22: word search game
  - # 32: graphical analysis of pseudorandom
  - # 37: matrix library
  - # 43: encryption algorithms

TRINITY WESTERN UNIVERSITY