

Review: lectures 1-17

16 Feb 2009

CMPT166

Dr. Sean Ho

Trinity Western University

Object-oriented concepts

- Everything is an object: attribs, methods
- A program is a set of objects passing messages
- Each object has local storage (composition)
- Every object has a type (class)
- Objects of same type can receive same messages

- UML: class diagram, use-case diagram, design patterns

Developing in C++

- Code (*.cpp) vs. headers (*.h)
 - The C pre-processor, #include, <> vs. ""
 - Comments and doc-comments
- Integrated development environments
- Compiling: object files (*.o)
- Linking: producing executable

C basics

- 8 built-in types
- if/else, switch/case
- while, do/while, for
 - break, continue
- Operators: + - * / % and or not << >> & | ^
 - i++ vs. ++i
- cin/cout
- Arrays: declaring, initializing, accessing
- extern, 2 meanings of static

Writing classes

- public / private / protected
- Subclassing
- Constructor, destructor
 - Default parameters
- Constructor initializer list
 - Calling the superclass constructor
- Overloading operators

- Class variables/methods (static)

Inheritance

- Designing class hierarchies: component design
- “Has a” vs. “is a kind of” vs. “knows how to”
- Virtual functions and polymorphism
- References (cf pointers?)
- Copy constructor
- Pure virtual functions and abstract superclasses
- Abstract superclasses vs. method interfaces

Namespaces

- Creating namespaces, accessing
 - two uses of 'using'
- The default anonymous/unnamed namespace
- Namespaces vs. 'static'

File I/O

- `<fstream>`
- `ifstream`, `ofstream`
- `getline()`
- `<vector>`
- `<string>`: initializing, operations, `find()/replace()`, `length()/capacity()/reserve()`, lexicographic sorting

Exceptions

- try/throw/catch
- Accessing the caught exception (getting auxiliary data)
- Standard exception classes: `<stdexcept>`, `runtime_error`, `logic_error`
- Designing exception class hierarchies

GUIs and FLTK

- Highlights from history of GUI
- FLTK basics