Semester review

14 April 2008
CMPT166
Dr. Sean Ho
Trinity Western University

TRINITY WESTERN UNIVERSITY

CMPT166 Course Overview

- Object-oriented programming concepts
- Design patterns
- Java language constructs
- Interfaces and the Java event model
- Swing
- File and stream I/O
- TCP/IP networking
- Multi-threading
- Generics

Object-oriented programming

- Program organization: classes, instances
  - methods, instance variables
  - program flow: main(), constructor
  - UML class diagram
- Inheritance
  - Designing inheritance hierarchies: UML
  - Overriding methods; polymorphism
- Access control
  - private, (package), protected, public

Design patterns

- Model–View–Controller
- Creational patterns:
  - factory, prototype, singleton
- Structural patterns:
  - facade, adapter, proxy
- Behavioural patterns:
  - observer, mediator, memento, command
- Collections and iterators

Java syntax constructs

- JVM/JDK, compiling, naming, packages, jar
- Primitive types
- String, Math libraries
- if, while, for
- Method overloading
- Arrays
  - contrast: Java arrays, Java ArrayList, Python list
- static
- final: variables, methods, classes

Interfaces and the event model

- Interfaces
- Abstract classes
  - Abstract methods
  - Compare with interfaces: Collection, Set, List
    - AbstractCollection, AbstractSet, AbstractList
- Event model
  - ActionListener, actionPerformed(), ActionEvent
  - Delegate classes for event handling

Swing

- Outline of a basic Swing program
  - main(), createAndShowGUI()
  - JPanel/JFrame constructor (widgets, layout)
  - Event handler: actionPerformed()
- Widgets: JLabel, JButton, JTextField, JTextArea
- Event interfaces *Listener: Action, Item, Mouse, MouseMotion, Window, Key
- Graphics: graphics context, clip, paint()/paintComponent(), drawLine(), drawRect(), drawArc()

File and stream I/O

- I/O streams:
  - FileInputStream (bytes), FileReader (text)
  - ObjectInputStream (object-based)
  - buffering, flush()
- RandomAccessFile: seek()
- Formatted text I/O: Formatter, Scanner
- Serialization
  - Serializable interface, transient
  - optional readObject()/writeObject() methods

TCP Networking

- Client vs. server; TCP vs. UDP; hosts, ports
- TCP server: ServerSocket
  - TCP socket connection: Socket
- Outline of TCP server:
  - new ServerSocket(*port*)
  - accept() (blocking) (returns a Socket)
  - getInputStream() / getOutputStream()
- Outline of TCP client:
  - new Socket(*host*, *port*)
  - getInputStream() / getOutputStream()

Multi-threading

- Spawning, parent-child model
  - Runnable interface, Thread class, run() method
- How to use threads to keep GUI interactive
- How to use threads in a TCP server
- Java Executor vs. OS task scheduler
  - Thread vs. task
- Synchronization, mutex, locks
  - Java synchronized methods

Generics

- Declaring a generic class: type parameter
  - Using the type parameter in methods
- Instantiating a generic class
- Multiple type parameters
- Constraints on type parameters
  - Superclass, interfaces
- Inheritance and generics
- ArrayList example