Final Exam Review Topics
Fall 2017

Exam technical details

- When: Thursday, December 14th @ 8am (!)
- Seats are assigned
- Up to two pages of own notes allowed
  - 8.5x11 paper (double sided is fine). Typed or handwritten.
- No electronic devices
  - Including calculators, watches, iwatches, phones, laptops, tablets, …
- Mostly multiple choice
  - Ethics problems will largely be free-answer
- Can grade multiple choice as you exit the exam

Topics from Exams 1&2 Are Still In Play

Exam 1:

- Basic OOP
- Inheritance
- Polymorphism
- Overriding vs overloading
- Abstract classes and interfaces
  - Comparable vs Comparator

Topics from Exams 1&2 Are Still In Play

Exam 2:

- Generics
- Java Collections Framework
- Enums
- Graphical User Interfaces
  - JFrame, JPanel, JButton, JRadioButton, Jlabel, JTextField
  - Layout managers

Graphics

Low-level mechanisms that allow a Component to draw itself

- Basics of the Graphics class: drawing lines, curves and strings; defining color
- Basics of the Graphics2 class: shapes

- repaint(): ask the graphics system to schedule a component for update
- paintComponent(): method called by the graphics system to actually do the painting

Event-Driven Programming

- Events vs Event Listeners vs Event Generators

- Button presses (Action events)
- Keyboard events
- Mouse events
- Timers

High-Level GUI Components & Related Classes

- JScrollPane
- JList
- JMenu
- Pop-up dialog boxes
- FileChooser

- DefaultListModel & data models in general

Files and Streams

Know the basic concepts

- FileXStream
- BufferedXStream
- DataXStream
- ObjectXStream

Recursion

- Breaking big problems into smaller problems
- Defining base and recursive cases
- Recursion vs iteration

Recursion Notes

- Method calls require a certain amount of "overhead"
  - Time and memory
- Any algorithm that is implemented as a loop can also be implemented recursively
  - Would we want to do this?

Recursion Notes

How about the other way around: can any recursive algorithm be implemented with a loop?

Recursion Notes

How about the other way around: can any recursive algorithm be implemented with a loop?

- Yes: but you would also need a stack data structure to keep track of all of the work left to do
- Note that method calls ***are*** stack operations

Efficiency Choices

Loop versus recursion

- The choice comes down to your specific situation
- In general:
  - Loops are more efficient with respect to time and memory, but may need more work to get it right
  - Recursion is often more elegant, but can cost time and memory

Ethics

- Basics of the different ethical theories
- Ethical principles & their sources
- Apply ethical principles and rules to novel problems of privacy and property

Same form as the labs (may have some multiple choice questions, too)