Outline

• Final class
• Advanced topics
  – Finish off some stuff from last class
• Review and overview
  – What we covered and how it fits together
  – Requested topics
  – Chance to practice on last semesters adopted exam….
  – Also will post last semesters sample final to look at…
    • Although course has been adopted somewhat
Announcements

• Please start early on studying
  – Can help you during office hours
  – TAs are around
• Look over class notes, assignments, midterms, and sample code
  – Understand them
  – Now is the time to review homework grades
• Open notes/book exam

Screen shot

• Anyone know how to generate a screenshot of your Othello game?
Eclipse question

• Hope you have tried to use eclipse

• Ctrl + space ??

• Ctrl + mouseclick ??

Review outline

• High level review
• Low level
• Actual exam
Most Intro Programming

Focus on teaching programming skills not so much on understanding how programming language and skills meet:

• Start
• Compute something
• End

Real programs

• So how do you deal with the following parameters?
  • Always on
  • Multiple users
  • Multiple states
  • Shared resources
  • Flexibility
  • Cost involvement
  • Outside errors
Take away lessons

• To really program:

• Need to understand how to solve problem correctly
• Need to understand how to design the program
  – OOP approaches
  – Programming language choices
  – Project requirements
• Now you can write a program!

Design issues we have covered

• UML
• Class design
• Understanding how class design diagram translates into real program
• How to divide work between objects
• Design patterns
• Design of objects
• Choice of logic representation
• Implementing your ideas in SWING
  – Knowing what to choose
  – Knowing how to use them
  – Knowing how to debug
Patterns should know about

- Observer
- Iterator
- Strategy
- Composite
- Decorator
- Stream

Concepts

- Polymorphism
- Inheritance
- Encapsulation
- Interfaces
- Iterators
- Anonymous class
- AWT Graphic obj
- Coordinate system
- Regular expressions
- Erasure
- Mouselistener
- mouseEvent
- ActionListener
- HashTable
- One way hashing
- Md5
- Shallow copy
- Class Reflection
- Generics
- Wild card generics
Concepts II

- Overriding
- Overloading
- Timer class
- Thread
- == vs .equals
- Autoboxing
- Enum type
- instanceof
- getClass() i.e. class obj
- Serializable
- Transient
- Abstract class
- Primitive types:
  - Class types
  - Interface types
  - Array types
  - The null type
  - .equals rules

Concepts III

- Kth highest number
- Maze searching
- Jar tool
- Javabeans
- Threads
- Locks
- Synchronization
- Sleep
- Comparable
- Comparator
- Scheduling algorithms
  - Round robin
  - FCFS
  - SJF – shortest job first
  - JProgressbar
  - JPanel
  - Race condition
  - Graphics2D
  - Context switching
Book

- Make sure you are familiar with chapters
  - 5
    - patterns
  - 6
    - Objects and inheritance, graphical stuff
  - 7
    - Object model and object class, generics, reflection etc
  - 9
    - Threading model

Random question?

- Why is it when we put up a graphic window we call setVisible last?
practice

• Practice exam

Something different

• I meant to do this at the end of last class

• Was giving you an overview of how to think…

• Caution: none of the following examples are meant to be taken literally, it is meant only to educate you on object-oriented design
Password Guesser

- Say you have a word document, password protected

- Any ideas on how to get in to read the document?

- Assuming:
  - Can enter as many passwords as you want up to three per one machine
  - More than three in under a minute, document self destructs
    - If I designed it, it would take your hard drive along 😊
  - As long as they are entered more than 1 minute apart, they don’t count as part of three

OOD

- Need to break it up into pieces

- Need to outline requirements

- Any ideas?
Requirements?

• Communication system
• Password generation system
• Multi threaded environment
• Multi instance environment
• Need to think in advance

Example 2

• How would you design a better protected document?
Question

• How would you program a search engine?

Next step

• DON’T panic
• Finish any homework you have left
  – Get help if you need it
• Get your notes together
• Start studying
• Email questions or come by OH if you need it
• Be awake during exam!