WHAT IS THIS COURSE ABOUT?

• Short Answer:
  • OOP mastery
  • No more toys
  • Plan, then do (design, then code)
  • Student to Pro
The LONG... answer = Software Development Lifecycle

- The *methodology* for constructing software systems of high quality.

- What properties make a software system high quality?
  - correctness
  - efficiency
  - ease of use (by other programmers in the case of frameworks)
  - reliability/robustness
  - maintainability
  - modifiability
  - extensibility
  - scalability
Klocs (1,000s Source lines of code)

- As programs get larger, these goals become much more difficult to achieve. Why?
  - program complexity
  - team complexity
As programs get larger, these become much more difficult to achieve.

- program complexity
- team complexity (more people are involved)

**How can these properties be achieved?**

- By using well proven, established processes
  - preferably while taking advantage of good tools
Software Development Lifecycle

- Other Steps to Consider:
  - Software Integration:
    - Done in large projects
    - Combine developed software into a cohesive unit

- Software Maintenance:
  - Follows Deployment
  - Monitoring and Updating deployed software
Software maintenance

- Follows Deployment

- Monitoring and Updating deployed software
Updated Software Development Lifecycle

- **Waterfall Model:**
  - Many variations:
    1. Requirements Analysis
    2. Design
    3. Evaluate Design
    4. Code
    5. Test, Debug, & Profile Components
    6. Integrate
    7. Test, Debug, & Profile Whole Program
    8. Deploy
    9. Maintain

- Note that there are many variations
Software Development Lifecycle

- There are other models:
  - Agile Programming
  - Extreme Programming
  - Pair Programming
  - Etc.

- We’ll talk more about these at the end of the semester

(c) Paul Fodor
Software Development Lifecycle

- Software Jobs:
  - Programmers = the most time consuming job in software development
  - Additionally, you should know *how to design, program, test, debug software*
    - Other types of jobs beside programmers:
      - Designer
      - Database, Network, Security Administrator
      - Tester
      - Project Leader
      - Manager
      - Documentation developer / Instructor
      - Founder/CEO

- NOTE: designers & programmers on a project may not be the same people!
Design, then develop

- We will design all classes before coding
  - not easy to do
- UML is used for software design
- You cannot design a system unless you really understand the necessary technology
  - designs cannot be created without testing
  - trying out different small-scale examples (HWs 2 & 3)
The HW Plan

• HW 1 – Build Process
• HW 2 – Technology Ramp-Up – GUls, Events & XML
• HW 3 – Technology Ramp-Up – 2D Graphics & Threads
• HW 4 - UML Design
• HW 5 – Implementation Stage #1
• HW 6 – Implementation Stage #2
• Final Project – Completed Work
What is a framework?

- More than just one class, but many classes working together
- Groups of classes that form the basis for customization
  - cooperating classes for a particular technology
    - ex: multimedia, the Web, databases, etc.
- used to build new applications & other frameworks
- Example: what’s Java’s application framework for the domain of GUI development?

Applications Using Frameworks:

- App1 calls methods of Framework objects
- Framework calls methods of App1 & App2 objects
- App2 calls methods of Framework objects
- Application #1
- Application #2

(c) Paul Fodor
Common Java Frameworks

1. Spring MVC
2. Struts
3. Apache Axis
4. Apache Xerces
5. Hibernate
6. JDOM
7. Java Applet
8. Apache Velocity
9. Apache ORO
10. JAX-WS

Framework developers must explain how to use them all together properly:
- API
- Tutorials

Frameworks are open source as well as for purchase.

Think about how you might create a framework.

Gaining the ability to make frameworks will make you a powerful developer.

Lots and lots of frameworks

- Google App Engine
- enchant.js
- jQuery
- AngularJS by Google
- Bootstrap
- Unity
Framework documentation

• Frameworks are many classes working together
• Framework developers must explain how to use them
  • API
  • Commenting
  • Tutorials
• Frameworks can be open source, free, proprietary
Who cares?

- We are constantly using Java frameworks
- Think about how you might create a framework
- Learning how and why to make frameworks will make you a powerful developer