January 28th, 2003

Course Goals

To make you a better programmer

- Deconstruct relevant programming problems
- Solve them in an object-oriented style, focusing on
  - Reusability
  - Maintainability (clarity)

Style

- Interaction
  - This is your course: what do you want to learn?
- Discussion
  - Not just professor/TA to student, but student to student, with regard to ideas, techniques, and solutions
- Learn by doing
  - If you don’t put effort into the programming projects, you will learn very little

Approach

- Methods and styles of (sequential) object-oriented (OO) programming.
  - abstraction, particularly in the use of interfaces and design patterns to improve reusability and reliability
  - will use the Java programming language exclusively
  - homework projects will build on each other
- Principles and practice of concurrent OO programming.
Topics

• Java (end of next week)
• Programming techniques and tools
• Object-oriented design for sequential programs
  – OO principles
• Design patterns
• Concurrency
  – concurrent programming in Java
  – design patterns
• To be determined
  – perhaps distributed programming in Java

Textbooks

• Primary Texts
  – Barbara Liskov and John Guttag, Program Development in Java
  – Bruce Eckel, Thinking in Java (3rd Edition), Prentice Hall, 2002
    • Java primer
    • A complete copy of the book can be downloaded for free
• See web page for more useful resources

Class Accounts

• We will have accounts on CSIC machines
  – Linux cluster; RedHat 7.3-based
  – Lab is in room 3107 CSIC
• Class accounts will be emailed
  – to the email account registered with UMEG
  • so make sure you check or forward that account!
  – if you don’t receive an account by Thursday, talk to TA

Software

• Will be using Java 1.4
  – http://java.sun.com/j2se/1.4/docs/api/index.html
• May wish to use the Dr Java IDE
  – Installed on all CSIC machines
  – http://drjava.sourceforge.net/
• Will make use of JUnit testing package
  – Starting with hw #2, will include unit tests
Projects

- Focus on server applications
  - Encourages modular, abstract design
  - Admits natural use of concurrency and distribution
  - Relevant in our connected society
- Start small and build up
  - Develop a sophisticated family of software by course’s end

First Project

- Due Feb 12th
- Implement an extensible web server
- Web page is up describing the project

Project Submission

- Projects due at 6pm on due date
  - by Unix time of day
  - you must submit a good-faith effort
    - you can be failed for the course if you do not
- Use online submission procedure
  - Submit early and often
  - Can provide recovery from previous submit
  - Details provided later

Project Commentary

- After the projects are graded, you will be emailed two other (anonymized) submissions
  - respond with commentary on each with regard to the goals and techniques we are teaching
  - you will be graded on the usefulness of your commentary
Tentative Grading Plan

<table>
<thead>
<tr>
<th></th>
<th>#</th>
<th>% each</th>
<th>% total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Projects</td>
<td>6</td>
<td>8.33</td>
<td>50</td>
</tr>
<tr>
<td>Mid-terms</td>
<td>1</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>Final</td>
<td>1</td>
<td>30</td>
<td>30</td>
</tr>
</tbody>
</table>

Discussion and Questions

- Read the newsgroup – csd.cmsc433
  - Only visible from inside UMD
  - For class discussion
    - TA’s will read regularly, but may or may not respond. Do not expect real-time responses. This is not a substitute for coming to office hours.
    - NIT: As in the rest of life, don’t believe everything you read!
- Don’t cross the line
  - know the academic integrity procedure and follow it (see web page for more)

Office hours

- Professor Bill Pugh (pugh@cs.umd.edu)
  - M 11am-12n, TuTh 3:15pm - 4:00pm
  - Or by appointment
- TA: Sasan Dashtinezhad
- Always posted on class webpage:
- Can check with Professor or TA for other section
  - short questions only

Stay up to date


Contains:
- Lecture notes
- Project assignments
- Resources
- And more!