CMSC433, Spring 2004
Programming Language Technologies and Paradigms
http://www.cs.umd.edu/class/spring2004/cmsc433

Jeff Foster

January 29, 2004

Course Goal

To make you a better programmer

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

Approach

- Will use the Java programming language exclusively
  - But the ideas apply to other languages equally well
- Sequential object-oriented (OO) programming
  - Basic principles and tools, and
  - *Design patterns* to improve reusability and reliability
- Concurrent OO programming
  - Shared-memory multi-threading (Java Thread class), and
  - Distributed message passing (Java Remote Method Invocation)

The Need for Abstraction

Fragment 1:

```
found = false;
for (int i = 0; i < a.length; i++)
    if (a[i] == e) {
        z = i;
        found = true;
    }
```

Fragment 2:

```
found = false;
for (int i = a.length-1; i >= 0; i--)
    if (a[i] == e) {
        z = i;
        found = true;
    }
```


Abstraction

- A better abstraction may be

```
found = a.isIn(e);
if (found)
    z = a.indexOf(e);
```

- Improves
  - Reusability
  - Maintainability
- Abstraction is the key to good design

Topics

- Java review (2-3 lectures)
- Programming techniques and tools
  - Specifications and testing
- Design patterns
- Refactoring
- Concurrency
  - Concurrent programming in Java
  - Design patterns
  - Distributed programming
- To be determined
  - Possibilities include security, event-based programming, reflection, memory management, ...?

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

Textbooks

- Primary Texts
  - Barbara Liskov and John Guttag, Program Development in Java
  - Bruce Eckel, Thinking in Java (3rd Edition)
    - 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 9
  - Lab is in room 3107 CSIC
- Class accounts will be emailed Friday morning
  - To the email account registered with UMEG
    - So make sure you check or forward that account!
    - Right after class: **check your e-mail address on TESTUDO**
  - If you don't get an account by Friday, e-mail us
- You may work on any machine you like, but...
  - Make sure you code runs on the linux lab

Software

- Will be using **Java 1.4.2**
  - http://java.sun.com/j2se/1.4.2/docs/api
- May wish to use the **Dr Java** IDE or **Eclipse** IDE
  - Installed on all CSIC machines
  - http://drjava.sourceforge.net/
  - http://www.eclipse.org
- Will make use of **JUnit** testing package
  - Part of hw #2, may want to use later yourself
  - http://www.junit.org

Projects

- Focus on networked 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

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
  - Late submission up to 9am the next morning
    - Score is multiplied by 0.9
- Online submission procedure (details later)
  - Submit early and often
  - E-mail course staff to recover previous submit
    - Don't rely on us, though – back up your own work

More on Projects

- Your programs **must** compile
- Grading will generally be automated
  - Occasionally a single bug will cause a lot of test cases to fail; let us know if this is the case for you
- Projects are the key to this course
  - Start early
  - Ask questions on newsgroup or in office hours

Project Commentary

- For some projects, 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

Exams

- Midterm: Thursday, March 18
  - Just before spring break
- Final: (Unofficial) Wednesday, May 19
  - Covers all of course
  - But roughly 2/3 new material, 1/3 old material
- Do **not** schedule travel for these dates!
  - I will be jealous, not sympathetic

Tentative Grading Plan

	#	% each	% total
Projects	6	8.33	50
Mid-terms	1	20	20
Final	1	30	30

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.
    - NB: 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 Jeff Foster, jfoster at cs.umd.edu
  - 4129 AVW
  - M 1:30-2pm, TuTh 10:30 - 11:30am
  - Or by appointment
- TAs: Mujtaba Ali and David Greenfieldboyce
  - Office hours in Linuxlab
  - MW 10am-12pm, W 3-5pm, F 12-1, 4-5pm
- Always posted on class webpage:
  - http://www.cs.umd.edu/class/spring2004

Stay up to date

http://www.cs.umd.edu/class/spring2004/cmsc433

Contains:

- Announcements
- Lecture notes
- Project assignments
- Resources
- And more!