COMP 303B - Programming Techniques

Laurie Hendren

Winter 2005

Official Course Catalog Description

Software architecture, design patterns, object-oriented programming concepts, profiling and optimization. Students will implement a significant programming project.
Prerequisites: COMP 206, COMP 251, COMP 302
4.000 Credit Hours

General Information

Course Web Page: `http://www.sable.mcgill.ca/~hendren/303/`

Time: Tuesday and Thursday, 2:35-3:55

Place: Trottier Building 0070

Instructor:

Laurie Hendren
Office: McConnell 228
Office Hours: Monday and Wednesday, 11:30-12:30
E-mail: hendren AT cs.mcgill.ca

TA:

Imran Majid
Office: McConnell 234
Office Hours: Monday, 12:00 - 13:00 and Thursday, 13:00 - 14:00
E-mail: imajid AT po-box.mcgill.ca

Detailed Course Description

This course is intended to provide an opportunity for students to learn tools and techniques for developing large object-oriented (OOP) and aspect-oriented programs (AOP). The first half of the course will concentrate on object-oriented programming using Java as the application language and the second half of the course will introduce a new aspect-oriented programming language, AspectJ.

An important theme of the course is that students will put what they are learning in lectures into practice by developing a term-long course project. This year the project will based on the the

ICFP 2004 (International Conference on Functional Programming) Programming Contest. The original contest rules may be found at:
`http://www.cis.upenn.edu/proj/plclub/contest/index.php`.

This contest involves an "ant" world and a contest between ants from different teams. Our project will have multiple milestones including an initial design and implementation of a command-line simulator for the ants, a GUI visualizer for the simulator, a driver program to run a competition between ants, and the design of your own ant. We will start in early milestones using only Java, but in the second half of the course we will also integrate the use of AspectJ.

Another important theme of the course is that we want to make use of freely available software development tools including (but not limited to) javadoc, subversion, ant (the tool, not the project), JUnit, UML tools and the abc AspectJ compiler.

In addition to the project there will also be several smaller assignments which will allow you to practice with concepts covered in the lectures.

Required Textbook and Reference Material

The required textbook is *Object-Oriented Design and Patterns* by Cay Horstmann. The textbook is available in Paragraphe bookstore located on McGill College Avenue. Additional online resources for the book can be found at:
`http://www.horstmann.com/design_and_patterns.html`.

There are three useful reference books about Java, all published as part of the Java Series published by Addison Wesley (refer to `http://java.sun.com/docs/books/`). The titles and authors are:

- *The Java Virtual Machine Specification*, by Lindholm and Yellin.
- *The Java Programming Language*, by Arnold and Gosling.
- *The Java Language Specification*, by Gosling, Joy and Steele.

A complete list of reference material will be listed on the course web page. Please refer to this page often to see the appropriate reading. Refer to the course web page for more details on the exact readings for each week.

Evaluation

Your performance in the course will be evaluated as a combination of a written/programming assignments, and a course project, a midterm and a final examination.

The final grade is calculated as follows:

- Assignments 15%
- Course Project (split into several milestones) 40%
- Meeting Milestone and Assignment Deadlines 5%
- Midterm 10%

- Final Exam 30%

Homework Assignment Policy:

Assignments should be handed in at the beginning of class on the day they are due. Work should be done individually, all material used in preparation of assignments and the project must be attributed to the appropriate sources. Late assignments, project milestones or final project reports will result in a lower mark for the meeting of deadlines/milestones.

Academic Integrity at McGill University

McGill University values academic integrity. Therefore all students must understand the meaning and consequences of cheating, plagiarism and other academic offences under the Code of Student Conduct and Disciplinary Procedures (see `http://www.mcgill.ca/integrity/` for more information). A helpful article on how to avoid plagiarism can be found at: `http://www.mcgill.ca/integrity/studentguide/` .

In terms of this course, part of your responsibility is to ensure that you put the name of the author on all code that is submitted. By putting your name on the code you are indicating that it is completely your own work. If you use some third-party code you must have permission to use it and you must clearly indicate the source of the code.