

University of Massachusetts Lowell
CS 91.204 Computing IV:
Advanced OO Programming and Software Tools

Professor Li Xu

Spring 2006

Course Description

The 91.204 Computing IV introduces students to the state of the art of object-oriented program design and the techniques and tools used in modern software development. The course will survey a broad range of subjects in software design and development. The major topics include: object-oriented analysis and design, OO design patterns, modeling using UML (Unified Modeling Language), IDE tools, build and debugging, testing, and documentation. Another focus of the course is to teach students the fundamentals of compiler/language processor design using these OO programming techniques. Topics include basics of language processing theory, regular expression, context-free grammar, intermediate representation and code generation. Students will get hands-on experiences by building a working compiler for the Chirp language running on Cricket educational robot. In the last subject block, we will cover important application-level Internet and Web technologies including TCP Sockets, HTTP, HTML and XML. The course will take a hands-on tool-based approach: students will learn the software methods and techniques through writing programs and using specialized software tools.

The course consists of lectures, homework assignments, exams, and small to medium-size programming projects which will emphasize object-oriented design and development methodology. The goal of this course is to provide a good understanding of principles and techniques in modern OO software design and effective use of modern software tools in software development. After successfully finishing this course, students are expected to develop the following skill set:

- understand and use OOAD design and methodology in program design and implementation
- understand and apply common design patterns
- understand UML modeling and draw UML diagrams
- understand unit testing and write test cases
- use software tools to perform program version control, build, debugging, testing and documentation
- use modern IDE framework and integrated development tools
- write event-driven programs
- understand lexing and parsing, build language scanners and parsers using automatic tools
- understand compiler intermediate representation and code generation through tree walking, build a simple compiler back-end
- understand TCP/IP, HTTP, HTML and XML Internet and web technologies and write programs using TCP sockets.

Course Prerequisites

CS 91.201 Computing III or equivalent. Students should be familiar with basic OO concepts and can program in one of the major OO languages, such as C++ and Java. All programming assignments in this course will be done in the Java programming language; however, prior knowledge of Java is not mandatory and we will introduce the Java language during the course.

Course Info

Instructor:

Professor Li Xu, xu@cs.uml.edu

Olsen 218, 978-934-1941

Course page: <http://www.cs.uml.edu/~xu/cs204>

Office hours and TA assignment will be announced on the course web page. You can also contact me for appointment through email. If you want to be sure I am in my office, use email to contact me.

Required Text:

Object-Oriented Software Development Using Java by Xiaoping Jia, 2nd Edition, Addison Wesley, 2003.

Head First Design Patterns by Elisabeth Freeman, Eric Freeman, Bert Bates, Kathy Sierra, O'Reilly, 2004.

As we will cover a wide array of subjects, there is no single book that covers all the course materials. Instead, we will use the above books for OO design and Java tutorial subjects and provide handouts and other references for the remaining topics. The related book sections will be assigned as reading homeworks.

The following two books are recommended as Java references: *Core Java 2, Volume I and II* by C. Horstmann and G. Cornell, 7th Edition, Prentice Hall, 2004. The 7th Edition covers the Java 5 features including generics and the new generics-based collection framework which will be covered in the course.

Assignments:

Usually, homeworks are in the form of programming assignments. All programming assignments will be done in Java. The assignments will count a total of 55 percent of your grade.

Exams:

There will be a midterm examination and a final examination that will count 40 percent of your grade.

Class Participation:

Students are expected to attend class meetings and participate in discussions and lab presentations. Class participation will count 5 percent of your grade.

Grading:

Your final score will be determined as follows:

Assignments: 55%

Midterm: 20%

Final Exam: 20%

Class Participation: 5%

Honor Code Policy: All class assignments are conducted under the University Honor Code. The honor code rules appear in the Graduate/Undergraduate Catalogs. For homeworks and programming projects, students are encouraged to consult each other, the TA, the instructor, or anyone else for that matter. However, the assistance offered or accepted should not go beyond a discussion of the problem and a sketch of a solution. You can use the following guideline: when it comes time for you to write your program or your homework paper, do not use any written material from the discussion. If you can reconstruct the discussion and complete the solution on your own, then you have learned the material (and that is the objective of this course).

Late Assignments, Regrading, and Absences from Exams:

Late assignments will have 10% of your raw score deducted for each 24-hour period for the first three days after the due date that the assignment is late. Late turn-in which is more than 3 days past the due date will be rejected without grading. This late policy will be strictly followed to ensure a fair learning environment for the course. Requests for regrades should be directed to the TA who originally graded the assignment. If you are still unhappy, see the instructor. Examinations will be given at alternate times only with appropriately documented medical excuses.