| **Lectures** | Crawford Hall, Room 223 | Mon., Wed., Fri. 1:00-1:50 PM |
| **Labs** | Tyler Hall, EnVision Lab (Rm 36) | Tuesday, 3:30-5:15 PM |
| **Professor** | Jean-Yves Hervé | Tel: 874 4400 |
| | | email: jyh@cs.uri.edu |
| | | Office hours: |
| | | Monday: 3:30 - 6:00 PM |
| | | Tuesday: 1:30 - 3:00 PM |
| | | or by appointment |
| **Teaching Assistant:** | Ben Roberts (labs) | Tyler Hall, Room 136 |
| | | email: ben@cs.uri.edu |
| | | tba |
| **Teaching Assistant:** | Pankaj Ahire (grading) | Tyler Hall, Room 128 |
| | | email: pahire@cs.uri.edu |
| | | tba |

**Equipment and References**

**Course text**

D.S. Malik  
*Java Programming: From Problem Analysis to Program Design*, 3rd edition  
Thompson, Boston, MA, 2008.  

There are way too many Java programming books on the market. Resists the urge to purchase additional books, thinking that this will make your life easier. It is much preferable for you to learn how to use properly the on-line documentation that comes with your development tools. Java is still moving and growing, fast. The fundamentals of the language have been kept mostly untouched, but new modules and functionalities are added all the time (communications, image processing, sound, etc.). Today Java is just too big for anyone to expect to master it completely. Instead, one should concentrate on mastering the fundamentals of the language and knowing where and how to find information on given topics.
Programming environment

We will be using the Eclipse integrated development environment (IDE) and the programming language Java. Eclipse was developed by IBM and is available as a free download for Windows 2000 & XP, Linux, and Mac OS X (http://www.eclipse.org/). The latest version of Eclipse is Version 3.3, which was released in August 2007.

The Course at a Glance

Prerequisites

MTH 111 (precalculus) or equivalent; prior experience with computers and programming.

Note that the word “Introductory” in the title of this course should not be understood as indicating that we will be starting “from scratch.” You are definitely expected to have done some programming in the not-so-distant past.

Computer Science was created as a subfield of applied mathematics and still requires a solid mathematical background. If your mathematical skills are weak you will run into problems sooner or later. This semester I will try to use some of the maths prerequisites in your laboratory assignments.

Brief Outline


Objectives

CSC 211 is intended primarily for Computer Science and Computer Engineering majors. During this semester, you will learn about programming, data representation, algorithm development, and object-oriented design. At the end of this course, you will be able to

- Analyze a simple programming problem and break it down into several subproblems or modules of programmable size;
- Develop an object-oriented design and algorithms for each module;
- Implement these modules in Java, test them, and debug them.

A comment

Programming is learned by (1) practice, (2) judicious use of reference manuals, (3) discussions with other programmers, (4) more practice.
Grading

The Mix

The following coefficients will be used to compute the final grade:

- Laboratory assignments 45%
- Final project 20%
- Midterms and quizzes 35%

Examinations

Two examinations (held during normal class hours) tentatively scheduled as follows:

- Midterm Exam 1: Wednesday, February 27,

Short quizzes will be held occasionally as well. Don’t expect me to announce the quizzes in advance because I typically don’t know myself that I will be giving a quiz until about 20 minutes before the class starts.

Laboratory assignments

You will not have to return anything immediately at the end of a lab: The lab time is there for you to experiment, ask questions (and hopefully get good answers), and generally trying to figure things out. At the beginning of the next lab, you will be asked to demonstrate the proper execution of your software and to hand in a printout of your code as well as a report on your work.

Final project

The final project will give you the opportunity to develop a piece of code more ambitious in size, and put into practice what you have learned over the semester. During the period reserved for the final project, there will not be new laboratory assignments. We will continue to have laboratory sessions, but these will be used to help you complete your project, with feedback from the lab assistant(s) and me.

Rules of Conduct

Late Submissions

All assignments should be turned in on the day and time they are due. If an assignment is not turned in on time, a 10% penalty will be applied for each late day (a fraction of a day will count as a whole day). Assignments more than one week late will not be graded.
If the assignment asks for multiple deliverables (code, project, report, Javadoc documentation) the date at which the last item was handed in will be used to determine the late penalty.

If you submit an assignment late, don’t just upload it to the server. You must also send an email to the TA (with Cc. to me) to inform her that the assignment is now complete and ready to be graded. You cannot just expect people to watch after you.

**Cheating**

Unless explicitly stated otherwise, all work should be done individually. Any evidence of cheating may result in expulsion from the class with a failing grade and will be brought to the attention of the Dean for disciplinary action.

You are strongly encouraged to discuss the assignments with other students, and try to figure them out together, but when comes the time of writing a report or developing code, you are expected to do it by yourself (or with other students from your group when the assignment explicitly mentions groups of 2 to 3 students).

Copying software from a book or from a web site without identifying the original author(s) of the work is **plagiarism**, which is a form of cheating.

**Laboratory use**

Any abuse of computer or software equipment will be brought to the attention of the appropriate authority for disciplinary action. Software piracy (the use, appropriation, or storage of illegal copies of software) is a form of abuse of the equipment and will be treated accordingly.

**Attendance**

Class attendance is not mandatory, although I strongly suggest that you make efforts to attend all classes. We will spend a significant part of the time in classes discussing about examples of programs, possible errors, bugs, etc. Don’t spend too much time taking notes during classes; rather make efforts to understand on-line what is going on. It will only get easier as we advance in the semester.

Lab attendance is mandatory and will factor in the evaluation.

**Tentative Class Schedule**

We are going to follow the order of chapters in the textbook pretty faithfully with the notable exception of topics related to GUI (Graphical User Interface) and graphics, which will not be taught in class, but rather learned ”on the job,” while doing lab and homework assignments.