CSC 211 – Introductory Programming and Design

Section 1 — Fall 2009

Lectures: Swan Hall, Room 305  Mon., Wed. 5:00-6:15 PM
Labs: Tyler Hall, EnVision Lab (Rm 036)  Tuesday, 3:30-5:15 PM
Professor: Jean-Yves Hervé
also: Tyler Hall, Room 104 (research lab)
Tel: 874 4400
email: jyh@cs.uri.edu
Office hours:
Monday: 3:00 - 4:30 PM
Tuesday: 5:30 - 7:00 PM
or by appointment

Teaching Assistants: Deborah Mathews
Tyler Hall, Room 135
email: dmathews@cs.uri.edu
Office hours: tba
Ben Roberts
Tyler Hall, Room 133
e-mail: ben@cs.uri.edu
Office hours:
Tuesday: 9:00 AM to 11:00 AM
Wednesday: 9:00 AM to 11:00 AM

Lab Assistant: Joel Barruos
n/a

Equipment and References

Course text
W. Savitch and F. Carrano
Java: Introduction to Problem Solving and Programming (5th Edition),

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, you should concentrate on mastering the fundamentals of the language and learning where and how to find information on given topics.
Programming environment

We will use 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 3.5 “Galileo”, which was released in June 2009.

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, October 14,
- Midterm Exam 2: Wednesday, November 18.

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 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, I will not assign 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).

The University Manual states:

*Students are expected to be honest in all academic work. A student’s name on any written work shall be regarded as assurance that the work is the result of the student’s own thought and study. Work should be stated in the student’s own words, properly attributed to its source. Students have an obligation to know how to quote, paraphrase, summarize, or reference the work of others with integrity. The following are examples of academic dishonesty.*

- Using material from published sources (print or electronic) without appropriate citation
- Claiming disproportionate credit for work not done independently
- Unauthorized possession or access to exams
- Unauthorized communication during exams
- Unauthorized use of another’s work or preparing work for another student
- Taking an exam for another student
- Altering or attempting to alter grades
- The use of notes or electronic devices to gain an unauthorized advantage during exams
- Fabricating or falsifying facts, data or references
- Facilitating or aiding another’s academic dishonesty
- Submitting the same paper for more than one course without prior approval from the instructors.
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. If you do not attend a lab session (or leave before the end of session evaluation) and fail to provide a satisfactory explanation for absence, you will get a grade of 0 for the lab, even if you turn in a perfect assignment.

Incomplete grade

In the past I have been much too lenient in giving an Incomplete grade at the end of the semester. This almost always eventually turns into a headache for me, so starting this semester, I am going to apply strictly the University guidelines:

A student shall receive a report of Incomplete in any course in which the course work has been passing up until the time of a documented precipitating incident or condition, but has not been completed because of illness or another reason which in the opinion of the instructor justifies the report. (Section 8.53.20 University Manual).

Note that overload from other courses does not fulfill the University policy conditions. Neither does a hard drive/laptop failure (we have a computer laboratory for your use).