Introduction to CS II – Data Structures
CAS CS 112 Spring 2010

(Syllabus)

- **Semester** Spring 2010
- **Teaching Staff**
  - Instructor – Hongwei Xi
    - Office Hours: TR: 5-6PM; W: 2-3PM
    - Location: MCS 172, e-mail: `hwxi@cs.bu.edu`
  - Teaching Fellow – Konstantin Voevodski
    - Office Hours: M: 12:30-2:00PM; R: 2-3:30PM
    - Location: PSY 223, e-mail: `kvodski@cs.bu.edu`
- **Lecture Times**: TR 12:30-2PM
- **Classroom**: GCB 209
- **Textbook**:
  - R. Sedgewick, Algorithms in Java (Parts 1-4), 3rd edition. Addison-Wesley, 2002, ISBN 0-201-36120-5.
  - R. Sedgewick, Algorithms in Java (Part 5), 3rd edition. Addison-Wesley, 2003, ISBN 0-201-36121-3.
- **Homepage**: `http://www.cs.bu.edu/~hwxi/academic/courses/CS112/Spring10`
- **Description**: This course starts by quickly revisiting, and then building upon, advanced programming concepts in Java taught at the end of CS 111. Then, the main focus of the course is on the design, analysis and implementation of fundamental data structures used throughout computer science. These include linked lists, stacks, queues, trees, hash tables, graphs, as well as specialized methods for searching and sorting. All of our implementations will be in the the object-oriented programming language Java. The emphasis in teaching this course centers around the following:
  - Developing elegant and efficient code from an abstract specification;
  - Literate programming (writing programs that can be read by humans as well as machines);
  - Developing a toolbox of advanced data structures for use in your future programming tasks, and an awareness of various design patterns that recur frequently in advanced programming;
  - Critical thinking about programs and the programming process, which involves:
    - Thinking about the best way to plan out the design using object-oriented design and appropriate features of Java;

    - Methodical and efficient development of the implementation using step-wise refinement and incremental testing and debugging (using appropriate debugging tools);
    - Being able to convince yourself of the correctness of the implementation by mathematical reasoning;
    - Analyzing the running time (efficiency) of programs by inspection and mathematical reasoning; and
    - Evaluating the efficiency and correctness of programs empirically, by using various tools in properly designed experiments.

- **Prerequisites**:

  This course is designed for students who already program with a CS 111 level of proficiency in Java. If you do not have significant previous exposure to programming, then you are requested to transfer to CS 111. You are expected to be familiar with UNIX and EMACS (or other equivalent text editor). Some help will be available in the section, but if you have not used UNIX or EMACS before, then you should attend the appropriate tutorials provided by B.U. Office of Information Technology.

- **Exams**:

  - First in-class midterm exam: Thursday, March 4, 2010
  - Second in-calss midterm exam: Thursday, April 15, 2010
  - Final Exam: 9-11am, Friday, May 7, 2010

- **Grades** The final score is calculated using the following formula.

  final score = 30%·(homework) + 30%·(midterm) + 30%·(final) + 10%·(attendance + participation)

  The final letter grade is calculated as follows.

  - **A**: final score is 85% or above
  - **B**: final score is 75% or above
  - **C**: final score is 65% or above
  - **D**: final score is 50% or above

- **Program Submission**: Program assignments are to be submitted via the *gsubmit* program. In case you need to learn how to use *gsubmit*, its documentation is available via a link on the homepage of this course.

- **Attendance**: It is expected that you will attend the lectures and the lab sessions for this course. I will take attendance at the beginning of half of the lectures. I also ask that you arrive in class on time as it is highly disruptive to have students flowing in throughout the class period. Moreover, when a student is a bordline case, I will check the attendance records before making a final determination.

- **Academic Integrity**: We adhere strictly to the standard BU guidelines for academic integrity. For this course, it is perfectly acceptable for you to discuss the general concepts and principles behind an assignment with other students. However, it is not proper, without prior authorization of the instructor, to arrive at collective solutions. In such a case, each student is expected to develop, write up and hand in an individual solution and, in doing so, gain a sufficient understanding of the problem so as to be able to explain it adequately to the instructor. Under *no* circumstances should a student copy, partly or wholly, the completed solution of another student.