17-630 Course Syllabus

Course Title: Data Structure and Algorithms for Engineers
Meeting Times: MW 3:00-4:20pm

Classroom: SCR 265
“...will, in fact, claim that the difference between a bad programmer and a good one is whether he considers his code or his data structures more important. Bad programmers worry about the code. Good programmers worry about data structures and their relationships.” - Linus Torvalds

Course Description:
The aim of this course is to provide students with a solid foundation in the concepts in the practical application of data structures and algorithms. This course emphasizes the practical application of techniques for writing and analyzing programs: data abstraction, algorithm verification, and analysis. This course will teach the students how to select and design data structures and algorithmic approaches that are appropriate for a given problem that they might encounter in practice. Students will learn practical techniques for reasoning about algorithmic correctness, computational complexities, performance, and various other properties and analyze trade-offs among various algorithmic strategies. This course provides students a broad mixture of theoretical knowledge complemented with numerous opportunities for practical hands-on experience to apply course concepts.

Course Objectives:
The primary objective of the course is to provide students with the skills they will need as practicing software engineers. The goal is to complement the programming experience that students already have formal underpinnings making them more capable, industrial strength software engineers. Specific Learning objectives include:

- knowledge of a broad class of data structures and algorithms commonly utilized in practice and when they might be utilized in practice
- the ability to design and evaluate data structures and algorithms for fitness of purpose
- the ability to decompose a problem into algorithmic parts and patterns
- the ability to select the appropriate design/algorithimic notations and representations to adequately describe and support reasoning about an algorithm and program design
- broad understanding of parallel algorithms and design concerns associated with concurrency
- knowledge of resource management

Prerequisites:
We assume that students know how to program and have written applications in at least one programming language.

Instructors:
- Mel Rosso-Llopart, Room SCR 270, Appointments by request.
- Anthony Lattanze, Room SCR
- Mathew Bass, Room SCR

Teaching Assistant:
(TBD)

Textbooks and Readings:
Required Texts

A variety of papers and URL readings will be provided in addition to these required textbooks.

**Brief Description of Course Assignments:**
This course includes numerous hands-on programming and analysis assignments. Students will program in a variety of languages. The programming assignments will be a combination of individual assignments and a team oriented final capstone project. In addition to programming assignments, students will be assigned readings to support the lecture materials. The course project will be completed in 2-3 person teams.

**General Grading:**

<table>
<thead>
<tr>
<th>Assignment</th>
<th>Percentage</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Individual Assignments</td>
<td>50%</td>
<td>These are small assignments that can vary: readings Q&amp;A, assignments based on lecture, coding, and so forth. These assignments will be done individually.</td>
</tr>
<tr>
<td>Final Capstone Project</td>
<td>40%</td>
<td>The course project will be completed in 2-3 person teams.</td>
</tr>
<tr>
<td>Instructor Judgement</td>
<td>10%</td>
<td>Participation in class, Team support, overall course effort.</td>
</tr>
</tbody>
</table>