# Syllabus

<table>
<thead>
<tr>
<th>Instructor</th>
<th>Pito Salas <a href="mailto:pitosalas@brandeis.edu">pitosalas@brandeis.edu</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>TBD</td>
</tr>
<tr>
<td>Office Hours</td>
<td>Go to this link to reserve an appointment: <a href="https://calendly.com/pitosalas/ftf">https://calendly.com/pitosalas/ftf</a></td>
</tr>
<tr>
<td>Class Times</td>
<td>Tuesday and Thursday, 3:30pm to 4:50pm</td>
</tr>
<tr>
<td></td>
<td>To be successful in this class it is strongly recommended that students attend the two hour recitation on Tuesdays 5:00pm-6:50pm</td>
</tr>
<tr>
<td>Location</td>
<td>TBD</td>
</tr>
<tr>
<td>Prerequisites</td>
<td>Cosi21a and Cosi166b (or permission)</td>
</tr>
<tr>
<td>Homework</td>
<td>Daily homework assignments; Major nanoTwitter project; Team assignments. Success in this 4 credit hour course is based on the expectation that students will spend a minimum of 9 hours of study time per week in preparation for class (readings, papers, discussion sections, preparation for exams, etc.).</td>
</tr>
</tbody>
</table>

## Course Description

How do you design a system that is meant to support millions of users? How will it perform? You certainly can’t just get a bigger and bigger computer. In this course we look at how to design and architect large scale web based systems. What are the classic algorithms and patterns used to achieve massive scale?

This is most certainly an evolving space with new work happening both in academia and industry. Yet there are well established and tested fundamentals of architecture which we will look at to better understand this fascinating and important space. We will look at caching, database partitioning, queueing, messaging and more. And we apply this learning working in teams of students to design and implement their own version of the Twitter backend from the ground up, and then stress test and measure it’s scalability using real world tools and technologies.

## Course Themes
1. **Architecture for Scale**: We also want to examine how to design systems which will scale under major load, whether it be many users, many transactions, lots of data, etc. Each of these are different kinds of scale challenges and will require different kinds of strategies. We will also of course look at generating synthetic load and measuring scale correctly.

2. **Software Design**: We want to look at programming beyond just ‘getting it to work.’ In large systems one thinks a lot about what happens to the program once its written. Software that’s used in the real world will always get revised, often. We will look at foundational principles of design such as the Single Responsibility Principle, Inheritance vs. Delegation tradeoffs, the importance of good naming, Test driven design, and many others.

3. **Major development project**: There will be multiple programming assignments where students will work individually or in teams to modify and enhance parts of the NanoTwitter software. Students will have to build and test on their own computers and then deploy software to servers in the cloud. We will subject the software to artificial loads to see how they behave and then see how changes affect performance.

**Experiential Learning**

This is an experiential learning course! You will be working in teams and discovering what that’s like, and how to be effective in that kind of setting. It’s something that will definitely come up in your future work, no matter what direction you take: business, academia, non profits, government, or entrepreneurship. You can count on learning from the challenges, obstacles and successes you encounter.

You will also be asked to be very self-reliant, figuring out things on your own, having to use the web to research tools and techniques to use, sorting through the noise and finding the best solutions. Everything will not be served up on a silver platter. You can count on developing your self confidence and perseverance and hopefully come to see the value of this in your future.

Overall you will be asked to think about what you are learning from these experiences, what you could apply in the future, and how this course may influence
the way you think about what you want to do next.

**Additional Information**

**Pre-requisites or permission**

Cosi21a and permission of the instructor are required for admission to this course. Permission is not required if you have taken Cosi166b. The reason for the prerequisite is that we jump very quickly into using a new language (Ruby) and framework (Sinatra.) If you are a good programmer and feel comfortable learning Ruby and Sinatra on your own, then I will be glad to give you permission. Please send me an email at pitosalas@brandeis.edu with your questions.

**Pre-work**

Students have all kinds of computers. I have nothing against Microsoft Windows, and you could take this course using just windows, but you’re life (and ours) is just going to be more difficult. It just seems that the open source tools and libraries we use don’t like Windows. So for this course, if your computer is set up with just windows, you are required to set up a Linux parition. It’s not hard. Here are the details: Setting up a Linux Parition

**Change Policy**

The instructor reserves the right to make changes to this syllabus and the associated curriculum web site if he deems it necessary. Any changes will either be announced in class or through e-mail. All students are responsible for finding out about such changes. Each student must be aware that not all assignments are listed in the syllabus. Students must use their common sense and not look for loopholes in the syllabus because, ultimately, the instructor has the final say in all matters. If you are confused on any assignment, ask the instructor for clarification.

By deciding to stay in this course, you are agreeing to all parts of this syllabus. In fairness to everyone, the syllabus must apply equally to all students without exception.
Crucial other Links

- Learning Objectives
- List of lectures
- Grading