CSE-416-03 25Fa Syllabus

Instructors

- Lecturer: Shuai Mu, shuai@cs.stonybrook.edu,
- Office hours: TBD.
  Note: Occasionally I would be occupied by other department affairs. Please check my calendar before you drop by to ensure I am available; you can also send a calendar invite (to the cs.sbu account, not the sbu account) to lock your slot.
- TA:
  TBD

Attendance Policy

- First-week attendance is mandatory, including those on the waitlist. Otherwise, you are at risk of being de-registered.
  - Stony Brook's first-week attendance policy
- After the first week, general attendance is not checked, which means you do not need to ask for my permission to take leaves for your other matters. However, being often absent could lead to an impression that you are not actively participating in the class, and then impact your grade. See the grading section for more information.

Goals

This class aims to prepare you for a software engineer job in modern tech companies, such as FAANG, or startup companies. This class emphasizes two types of abilities: hacking and communicating. We will achieve this mainly by PRACTICE. We will work out a project including its complete lifecycle: Design; Implement; Test; Improve; and Release.

Requirements

You are expected to have a "senior" standing, meaning you have completed most classes (U4 in Stony Brook's academic guidelines). Informally, you are expected to be "almost ready" to become a software engineer.

You have learned different types of programming languages. For example, a scripting language (no types) like Javascript, a strongly typed language, with GC (Java) or without GC (C/C++).

This class expects you to learn to use new languages of these types to finish simple tasks by using online materials without extra tutoring, which is a common expectation for software engineers.

No extra system prerequisite is required. However, our project can be considered by some as a “systems” project. We do not assume you already have the knowledge or you can learn it completely by yourself. We will go through the related concepts in lectures.

Project

We will build a file-sharing blockchain project to discuss important software engineering skills such as documentation, collaboration, code review, testing, etc.

There are two classic projects in the area, you can take a look at them for reference:

- IPFS:
  - ipfs.com
  - http://mpaxos.com/teaching/ds/23fa/readings/filecoin.pdf
  - http://mpaxos.com/teaching/ds/23fa/readings/ipfs.pdf
- ETH Swarm
  - https://www.ethswarm.org/

Thinking about the team buildup, a possible approach the divide the work is :

- Frontend (the client):
  - Web
  - Desktop
  - Mobile
- Backend (the “blockchain”)
  - DHT
  - Ledger
  - File sharing related functions.
- Application
  - E.g., a mega-like file share application, or vpn-like data service

Tech Stack

Regarding the toolchains (“tech stack”) selection, this part is also planned as a learning process in the class. You are expected to use any language that fits, not only the languages you already know. In principle, you are allowed to use any language and toolchains you deem reasonable,

as long as your team has a consensus. You are allowed to use any third-party languages or existing tools. But the main project is expected to heavily use two languages:

- Rust
- Typescript

The freedom of using any language/toolchain comes at a cost that you will have very limited tech support from the lecturer and TAs.

Teamwork

We will have a loose team structure. The class will have about ~10 teams. With 60 students attending, each team is expected to have 6 people, but with no hard limit. You are allowed to switch teams at any point you like.

Grading

THE GRADING IS ENTIRELY *SUBJECTIVE*. Please note that this evaluation method may differ significantly from conventional objective grading, such as examinations, to which you may be accustomed. However, it more closely reflects how performance is assessed in a professional employment setting. Given that this course aims to prepare students for real-world scenarios, I consider this experience more beneficial than any perceived inequity it may engender.

We will have an open review process to rank your performance to the whole class, updated approximately every month. Your letter grade will be distributed by:

A: top 30%
A-: 30-50%
B+: 50-70%
B: 70-80%
B-: 80-90%
C+: 90-?%

With the following exceptions:

- If your involvement is substantially below the average expectation of class participation (e.g., you are missing for half a semester, or if you are reported by your teammates for continuous absence), then you'll get an F.
- If your grade is close to the next level, you are allowed to do extra work to bump up your grade, but not if your performance is far away.

Exams

We will have no exams. We may have a few quizzes in class that do not count into your final grade.

Academic Integrity and Policies

Feel free to use *any* online materials/text/code, including GPT and other AI tools, as long as it is legal (e.g., you don't violate any open-source license) and is aligned with Stony Brook's official policy.

The following behaviors are subjected to penalties:

- Any dishonest behaviors, e.g., claiming credit for others' work
- Any malicious behaviors, e.g., sabotaging others' work

The standard penalty is a downgrade on your grade, severe violations will lead to an F.

Open Source Policy

All code will be open source under MIT or BSD license.

Schedule (Subject to change)

schedule (subject to changes):

Aug 25 Course Overview (Syllabus)
Aug 27 Lecture: Bitcoin
Sep 1 No class (Labor Day)
Sep 3 Lecture: Design Overview
Sep 8 Discussion: toolchain
Sep 10 Presentation: GUI
Sep 15 Presentation: GUI (Cont)
Sep 17 Progress review
Sep 22 Lecture: DHT
Sep 24 TA Demos
Sep 29 Progress review/UML
Oct 1 Presentation: Code Review
Oct 6 Workshop/Discussion: Design Doc
Oct 8 Progress review / report preview

Oct 13 No class (Fall Break)
Oct 15 Progress review / report preview (cont)
Oct 20 Lecture: Programming Patterns
Oct 22 Placeholder (TBD)
Oct 27 Progress Review
Oct 29 Presentation: Testing

Nov 3 Progress Review
Nov 5 Discussion: Boeing
Nov 10 Progress Review
Nov 12 Lecture: Relational Database
Nov 17 Progress Review
Nov 19 Presentation: Debugging
Nov 24 Presentation: Debugging (Cont)
Nov 26 No class (Thanksgiving Break)
Dec 1 Progress Review

Dec 3 Presentation: "Fake it until you make it"

Dec 8 Last day (TBD)

Stony Brook Calendar