

The University of Texas at El Paso
Google Tech Exchange – Spring 2023
CS 4390 - Software Development Studio

TechX 2023:

Software Development Studio Syllabus

Class Location:	Will be hosted virtually via Google Meet
# of Credits:	3 credits

1. General Course Description

1.1 Course Description

Software Development Studio integrates various components of students' undergraduate computer science curriculum, enabling students to work collaboratively on projects using professional tools and processes. This course bridges the gap between the academic and industry experience of software engineering.

1.2 Prerequisites

- Completion of two Computer Science courses at their university including Data Structures & Algorithms.
- Ability to solve coding problems using lists/arrays, dictionaries/maps, and graphs.
- Proficiency in a mainstream programming language such as C, C++, Go, Java, JavaScript, Python, or Rust.

2. Course Resources

2.1 Course Components

- **Lectures:** 1 hour 20 minutes, occurring 2 times per week.
- **In-class activities:** Group labs, discussion and in-class exercises during lectures.
 - TAs will be available to help students work through in-class activities.
- **Exams:** Midterm and Final covering the ideas and philosophies taught in class.
- **Group Project:** The main deliverable of the class, completed with a group of 3.
 - Project coaches will be available to assist and coach teams.
- **Individual Homework:** Practice for important skills taught in class.
 - Project coaches will be available to assist and coach individuals.

2.2 Course grade

We may elect to curve grades up, but we shall not curve grades down. The final course grade will be determined by your overall percent score in the course:

- A: [90, 100+]
- B: [80, 90)
- C: [70, 80)

- D: [60, 70)
- F: [0, 60)

Here is the breakdown:

- Mid-term: 5%
- Final Exam: 10%
- Class Participation: 5%
- In-class Activities: 10%
- Individual Homework: 30%
- Team Project: 40%

2.3 Asking Questions

If students have questions outside of class or lab, there are 4 venues where they can ask them:

- **Group chat:** TAs monitor [this Google Chat Space](#) for questions from students.
- **Email:** For private questions not appropriate for the group chat (e.g. a question that includes a snippet of their code), email the course staff at tx23-sds-staff@techexchange.in.
- **Office hours:** Sign up for office hours by selecting available slots [from here](#).
- **Directly email your instructors**

2.4 Required textbooks

None. Students will access assignments and readings via Google Classroom. Some useful resources that you may want to consult are:

- Cloud Shell: cloud.google.com/shell
- Git: lab.github.com
- Python: docs.python.org
- Flask: flask.palletsprojects.com
- HTML: www.w3schools.com

2.5 Department resources

The course will be co-taught with instructors from Google and faculty from partner universities.

3. Course Aims and Learning Objectives

3.1 Course aims

This course aims to teach students:

- The knowledge and skills to develop and deploy a full-stack web application using popular technologies and languages such as HTML, Python, Flask, and Firebase.
- How to work in a remote and collaborative environment that closely resembles the environment at real software companies using Git.
- Practice good code hygiene and testing.

3.2 Learning objectives

After completing this course, students will be able to:

- Understand existing code bases
- Utilize various APIs, frameworks, and libraries
- Design software engineering projects and create realistic milestones to track their progress
- Use version control systems (e.g. Git) to manage and contribute to a shared codebase
- Collaborate effectively within a team and practice concepts such as pair programming and agile development.
- Utilize industry standard software engineering techniques such as test driven development, code review, and design documentation.
- Communicate their designs to their project coaches and classmates

4. Sections

A M/W 5-6:20pm ET

- Instructor: Brian Noyama (they/he/any) noyama@techexchange.in
- LAs: Peter (he/him/his) paq@techexchange.in
Russell (he/him/they/them) raw@techexchange.in
Karen (she/her/hers) karenav@techexchange.in
- Project Coaches:

B M/W 7-8:20pm ET

- Instructor: Melissa Kohl (she/her/hers) meljkohl@techexchange.in
- LAs: Peter (he/him/his) paq@techexchange.in
Karen (she/her/hers) karenav@techexchange.in
Russell (he/him/they/them) raw@techexchange.in
- Project Coaches:

C T/Th 11am-12:20pm ET

- Instructor: Brian Noyama (they/he/any) noyama@techexchange.in
- LAs: Laura (she/her/hers) lauragilbert@techexchange.in
Nhu (she/her/hers) nhunguyen@techexchange.in
Juan (he/him/his) jmaraboli@techexchange.in
- Project Coaches:

D T/Th 3-4:20pm ET

- Instructor: Melissa Kohl (she/her/hers) meljkohl@techexchange.in
- LAs: Laura (she/her/hers) lauragilbert@techexchange.in
Nhu (she/her/hers) nhunguyen@techexchange.in
Juan (he/him/his) jmaraboli@techexchange.in
- Project Coaches:

4. Office Hours

Day	Time	Hosts
Monday	4-5pm ET 6-7pm ET	Brian Noyama, Karen Avner Melissa Kohl, Peter Amenewolde

Tuesday	10-11am ET 2-3pm ET	Brian Noyama, Juan Maraboli Melissa Kohl, Russell Webb
Wednesday	4-5pm ET 6-7pm ET	Brian Noyama, Russell Webb Melissa Kohl, Nhu Nguyen
Thursday	10-11am ET 2-3pm ET	Brian Noyama, Juan Maraboli Melissa Kohl, Peter Amenewolde
Friday	9-10am ET 3-4pm ET	Nhu Nguyen, Laura Gilbert Laura Gilbert, Karen Avner

5. Course Schedule

Date	Topic(s)	Homework/Activities
Unit 0: Python & Git		
Day 0: Jan 11 + 12	Intro, Syllabus & Project overview	Due: Pre-class survey Activity: Cloud Shell + Hello World
Jan 16 + 17	Martin Luther King Jr. Day	
Day 1: Jan 18 + 19	Intro to Git	Activity: Introducing Git + Git Conflicts Start: HW #1 Unit Testing
Day 2: Jan 23 + 24	Python Review	Activity: Wrapping up Intro to Git
Day 3: Jan 25 + 26	Pytest	Activity: Reviewing Python with Cloud Shell
Day 4: Jan 30 + 31	Testing Practices	Due: HW#1 Checkpoint Activity: Unit Tests
Day 5: Feb 1 + 2	General Software Testing	Activity: Integration Tests
Unit 1: Web Programming		
Day 6: Feb 6 + 7	Git'in Deeper Merge Conflicts	Due: HW#1 Unit Testing Activity: Git Conflicts
Day 7: Feb 8 + 9	Working on Projects + HTML	Start: Project #1 Flask Wiki Activity: Imposter Syndrome, Difficult Scenarios in Project Work
Day 8: Feb 13 + 14	Rest API + Intro to Flask	Activity: Calling a Web API, Flask
Day 9: Feb 15 + 16	Flask Testing + Flask-Login	Due: Project #1 Checkpoint Activity: Project Work Time
Feb 20 + 21	President's Day	
Unit 2: Engineering Process		
Day 10: Feb 22 + 23	Cloud + GCS Buckets	Activity: Project Work Time
Day 11: Feb 27 + 28	Requirements Engineering	Due: Project #1 Flask Wiki Start: HW #2 Wiki Features Design Doc

		Activity: Goal Modeling
Day 12: Mar 1 + 2	Midterm	
Day 13: Mar 6 + 7	Design Documents	Due: HW #2 Checkpoint Activity: Technical Diagrams
Day 14: Mar 8 + 9	Project Management	Due: HW #2 Wiki Features Design Doc Start: Project #2 Delegate work + Continuous Integration + Features Activity: Project Worktime
Mar 13-16	Spring Break	
Day 15: Mar 20 + 21	Software Quality (Continuous Integration, Environments, Testing)	Activity: Project Worktime (Continuous Integration)
Day 16: Mar 22 + 23	Merge Requests + Code Review	Activity: Review Example Code
Day 17: Mar 27 + 28	Software Ethics: ACM Code of Ethics, Security and Privacy	Due: Project #2 Checkpoint Activity: Ethics Scenarios
Day 18: Mar 29 + 30	Monitoring	Due: Project #2 Merge Requests Start: Project #2 Code Review for MR Activity: Design Monitoring
Day 19: Apr 3 + 4	Site Reliability Engineering	Due: Project #2 Code Review for MR Start: Project #2 Address MR feedback Activity: Design an SLO
Day 20: Apr 5 + 6	Retrospectives and Postmortems	Activity: Project Worktime (Deploy wiki to production)
Day 21: Apr 10 + 11	CSS, Accessibility and Inclusivity	Due: Project #2 Resolve feedback and merge MR Activity: Project Worktime (Add CSS to wiki)
Day 22: Apr 12 + 13	Presenting Technical Work	Start: Project #3 Final Presentation Activity: Difficult Workplace Scenarios
Day 23: Apr 17 + 18	Open Source and Licenses	Activity: Add a LICENSE to your wiki Activity: Push your wiki to a public Repo
Day 24: Apr 19 + 20	Final Exam	
Day 25: Apr 24 + 25	Final Presentations	Due: Project #3 Final Presentation Slides Activity: Final Presentations

6. Webcam Policy

We do **not** require students to turn on their webcam, and leaving your camera off will not affect your grade. However, students should use their webcam if able, and we will reach out individually to meet with students who consistently do not use their webcam.

7. Academic Integrity

7.3 Late Work policy

Late work will only be accepted with permission from the course instructor. Additionally, for late work students will have to **meet with their project coach** for one hour to grade their assignment together. If you require extra time, please email your instructor and project coach with:

- The reason for the extension
- The number of days you require

7.4 Attendance policy

Students are expected to attend classes regularly and on time, except for days noted as optional on the syllabus.

Students are required to attend class to complete the graded in-class activities. If a student requires an absence for an excused reason, they may email the instructor for the in-class activity and complete it outside of class.

7.5 Plagiarism policy

Use the table below to determine if you can collaborate on a given assignment.

What kind of help can I get on this assignment?

Type of work	Help allowed from
In-class activity	Lecturers, lecture assistants, or student enrolled in this class
Wiki Project	Project coaches, lecturers/assistants or students on the same team.
Homework assignments	Lecturers, lecture assistants or project coaches. Further collaboration permitted will be detailed in homework assignments.
Exams	No help permitted

7.6 Guidelines for Collaboration

Software engineering is inherently a collaborative process and in this class, we expect students to collaborate with **only** their fellow student(s) they are paired with in labs and grouped into for the unit projects.

The following are **not allowed**:

- Copy/pasting any code from anywhere other than from your assigned teammates, TAs and project coaches
 - This includes copy/pasting code snippets from online resources such as stackoverflow.com
- Sharing your code with other students that are not your assigned teammates.
- Reading code from other students that are not your assigned teammates.
- Doing your teammate's work for them.

The following are **allowed**:

- Asking and answering questions in the class Google Chat, as long as you **do not include multi-line code snippets**.
- Copy/pasting examples from any reference material (slides, practice problems, etc) distributed by lecturers/assistants in this class.
- Using any small code snippets that lecturers/assistants and project coaches share with students.
- Using open source software as packaged libraries (but please ask project coaches beforehand).

We should never see homework and project submissions that have plagiarized snippets of code. If that occurs, offending submissions will receive no credit.

When in doubt, reach out to project coaches or the instructors before using any code and include a comment in your submission explaining why.

In addition to this policy, all instances of plagiarism will be directed to the university administration, which will conduct the appropriate hearings.

8. ADA Policies and Procedures

If a student needs particular accommodations to be made, they must fill out [this form](#).