

---

---

---

# INTRODUCTION

WEEK 1 - 08/26/2019

Welcome to CSCI 3308 –  
Software Development Methods and Tools

# COURSE INFORMATION



Semester: Fall 2019



Credit: 3 credit hours



Lectures: Monday & Wednesday

Section 100: 10:00 a.m. – 10:50 a.m. in ECCR 150  
Section 200: 1:00 p.m. – 1:50 p.m. in GOLD A2B70  
Distance Section: All lectures will be available and viewable on the Canvas site.

# INSTRUCTOR

- Name: Sreesha Nath
- Email: [sreesha.nath@colorado.edu](mailto:sreesha.nath@colorado.edu)
- Office Location: ECOT 737
- Office Hours: MW – 3:00 pm – 5:00 pm. Please book appointment via Calendly to avoid extended waiting times.\*



\*Feel free to stop by any time for chocolates at my desk.

# TEACHING ASSISTANTS



Alecio Madrid  
[alma5914@colorado.edu](mailto:alma5914@colorado.edu)



Annika Muehlbradt  
[annika.muehlbradt@colorado.edu](mailto:annika.muehlbradt@colorado.edu)



Carl Mueller  
[carl.mueller@colorado.edu](mailto:carl.mueller@colorado.edu)

# TEACHING ASSISTANTS



Joshua Ladd  
[Joshua.Ladd@colorado.edu](mailto:Joshua.Ladd@colorado.edu)



Shreshtha Pankaj  
[Shreshtha.Pankaj@colorado.edu](mailto:Shreshtha.Pankaj@colorado.edu)



Yash Sapra  
[Yash.sapra@colorado.edu](mailto:Yash.sapra@colorado.edu)



Srinjita Bhaduri  
[srinjita.bhaduri@colorado.edu](mailto:srinjita.bhaduri@colorado.edu)

## ADDITIONAL SUPPORT STAFF

- Graduate Student Staff:
  - Karthik Siddaramanna - [Karthik.Siddaramanna@colorado.edu](mailto:Karthik.Siddaramanna@colorado.edu)
  - Janani Selvan - [Janani.Selvan@colorado.edu](mailto:Janani.Selvan@colorado.edu)
  - Venkat Arigela - [Venkat.Arigela@colorado.edu](mailto:Venkat.Arigela@colorado.edu)
- Course Assistants:
  - Jacob Malcy - [Jacob.Malcy@colorado.edu](mailto:Jacob.Malcy@colorado.edu)
  - Daniel Scott - [Daniel.Scott-I@colorado.edu](mailto:Daniel.Scott-I@colorado.edu)
  - Elliot Whitehead - [Elliot.Whitehead@colorado.edu](mailto:Elliot.Whitehead@colorado.edu)

# SYLLABUS

- <https://github.com/SreeshaNath/sreeshanath.github.io/blob/master/CSCI3308-syllabus.md>
- This syllabus includes detailed information about the course. Kindly read it thoroughly to avoid confusions.

# COURSE WEBSITE - CANVAS

- Course has been published on Canvas
- Please check the website regularly for updates on labs, assignments, reading materials, resources to ensure that no deadlines are missed
- All lecture videos will be available under the 'Lecture Access' tab on the menu on the left side of your course page.

# RECITATIONS

Section	Day & Time	Location	TA
101	W 5:00 p.m. - 6:40 p.m.	ECCE 141	Carl Mueller
102	F 8:00 a.m. - 9:40 a.m.	ECCR 235	Srinjita Bhaduri
111	Th 2:00 p.m. - 3:40 p.m.	ECCR 235	Shreshtha Pankaj
112	Th 4:00 p.m. - 5:40 p.m.	ECCR 235	Annika Muehlbradt
113	F 12:00 p.m. - 1:40 p.m.	ECCE 141	Alecio Madrid
201	W 6:00 p.m. - 7:40 p.m.	ECCR 235	Srinjita Bhaduri
202	Th 9:00 a.m. - 10:40 a.m.	ECCR 235	Josh Ladd
203	Th 3:30 p.m. - 5:10 p.m.	ECCE 141	Yash sapra
204	F 8:00 a.m. - 9:40 a.m.	ECCE 141	Josh Ladd
205	F 10:00 a.m. - 11:40 a.m.	ECCR 235	Shreshtha Pankaj
206	F 12:00 p.m. - 1:40 p.m.	ECCR 235	Annika Muehlbradt
207	F 2:00 p.m. - 3:40 p.m.	ECCR 235	Yash sapra

## NOTES ON WAITLISTS

- Class enrollments are constrained by the number of seats in a classroom.
- This class is constrained primarily by seats in the recitation sections, but also by the number of seats in the on-campus lectures
- The waitlist process is managed by software
- If a section is full, you go on the waitlist
- Positions in the waitlist move up and down depending on
  - Add/Drops
  - Year in School and Major
- Moving up in a waitlist
  - Preference is given to Seniors, then Juniors, etc.
  - Preference is given to Comp Sci majors, then others

# TEXTBOOKS AND READING MATERIAL



- No textbook for this course
- Reading material and other additional resources will be posted on Canvas every week.

## COURSE OBJECTIVES



- Learn the fundamentals of software development methods
- Gain exposure and practice using common industry tools that are likely to be used in the workplace
- Acquire useful skills for future programming classes
- Gain familiarity with common industry vocabulary for job applications and interviews
- Equip you to choose the best software tool for use in a specific situation
- Apply software knowledge and skills in a small group semester long project

# GRADING



Component	Percentage (%)	Points
Team surveys	2	20
Homework assignments	24	240
Exams	20	200
Labs	24	240
Team Project	30	300
<b>TOTAL</b>	<b>100</b>	<b>1000</b>

## LATE SUBMISSIONS



- You can receive a three-day extension on **any 2 homework or milestone assignments with a 10% grade penalty per day**. After 3 days, your assignment is considered past due and will not be accepted.
- Each Lab/Recitation includes a lab assignment. A predefined portion of the assignment for each lab should be submitted to the TA by the end of the lab. In some cases, you might need more time to complete the lab assignment. You can have up to 48 hours after the end of your recitation section to submit your work for that lab.
- In the event of a documented personal, family, or medical emergency, consult your TA about receiving a penalty free extension.
- If you know you will be missing a weekly lab recitation, you must talk with your TA before going to an alternate recitation period.
- If you attend an alternate lab session, then your lab assignment is due on the day you actually attend lab, not on the day of your normally scheduled lab recitation.

# HOMEWORK ASSIGNMENTS



The course includes **FOUR** homework assignments that comprise **24%** (240 points) toward your grade. Each assignment must be completed and submitted via Canvas by its due date to earn full credit.

- HTML & CSS (50 points)
- Java Script (50 points)
- SQL (60 points)
- REST API (80 points)

# EXAMS



- There will be TWO exams:
  - MidTerm - 1 during Week 8 Wednesday October 16
  - MidTerm - 2 during Week 13 Wednesday November 20
- Together the exams make up 20% (200 points) of your final grade
- The exams will be administered and graded by Canvas
- Each exam has a date/time window in Canvas, and will become unavailable in Canvas when that time is passed
- During lecture before each midterm we will have a review to help prepare you for the exam.

# GROUP PROJECT



- Groups of 5 or 6 students from the same recitation section.
- **GOAL:** Develop a working software application:
  - Frontend user interface, a backend database, layer(s) to connect
  - Document the design, development, testing process
  - Software development methodology and tools for the project
  - All documentation, code, and other materials stored in a GIT repository.
- Team Formation
  - Assigned by teaching staff
  - Survey
    - Work schedule, work habits, experience with various technologies

# GROUP PROJECT



- The final project grade for the team is determined by –
  - Progress towards product delivery
  - A working app
  - The difficulty of the project
  - The creativity and usefulness of the app
  - Group Presentation
  - Delivery of the materials required for each milestones 1, 2, 4, 5, 6, 7
- The final project grade for you as an individual is determined by –
  - Milestones 3 & 8
  - Your team's assessment of your contributions:
    - your time, your work, your attitude, your communication, your accessibility, your persistence

# PROJECT MILESTONES

Milestones	Points	Expectation
Milestone 1	40	Project Proposal
Milestone 2	35	Project Tools and Management Methodology
Milestone 3	50	1 on 1 student meetings
Milestone 4	45	Database design
Milestone 5	25	Application testing plan
Milestone 6	40	Project Presentations
Milestone 7	50	Final Project Report and Product Functionality
Milestone 8	15	Final Reflection



# GROUP PROJECT TIPS



- Do NOT wait to get started
- Meet early
- Meet often
- Team work is key
- Make your goal to develop a working app.
- Set the technical “bar” at a height commensurate with the overall technical skills of the team.
- Web site, mobile app, game, real-time data collection
- Deployed locally or in the cloud

# TOPICS

Unix Shell Scripting	Project Management
Waterfall, Agile development methodologies	Requirements Definition
Relational Database Design	Pair programming
HTML & CSS (front end)	NodeJS (middle layer)
SQL Query Language (back end)	Documenting your code
Web Services	Cloud Computing
Intellectual Property	Testing and QA
Static and Dynamic Code Analysis	Refactoring
Peer Review	Source code version control and managing conflicts
Application Security	Debugging your code

# COURSE SCOPE

- Many, many topics (wide)
- Variety of topics (technical, non-technical)
- Not enough time to go deep (shallow)
- “Simple and working” is better than “cool but not working”
- A wide breadth of student skills and experience
- You are NOT here to learn to code; you are here to learn to work together to deliver working software using some tools and skills that we will learn about
- Focused on “marketable skills” as seen by Hiring Managers

WE'RE HERE  
FOR YOU! 

NO LABS THIS WEEK!

