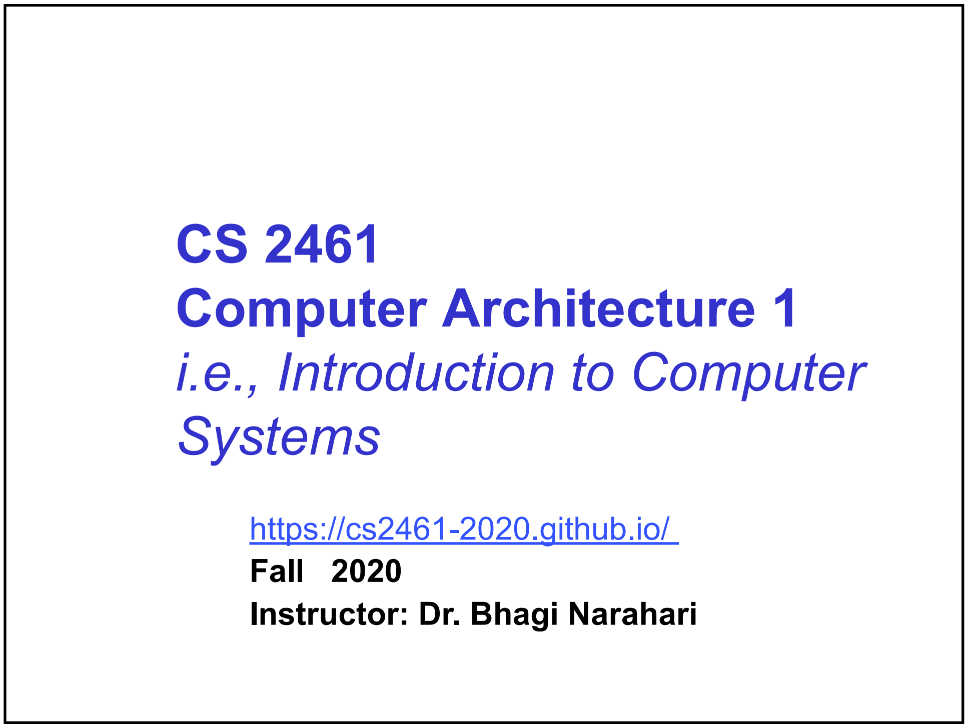
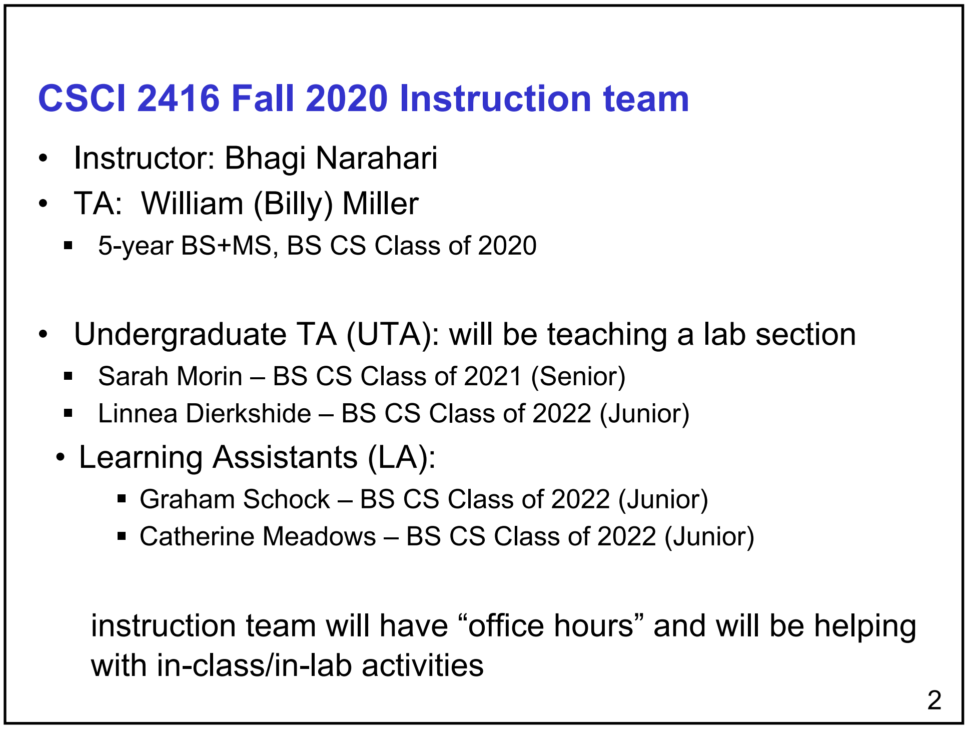


CS 2461
Computer Architecture 1
i.e., Introduction to Computer Systems

https://cs2461-2020.github.io/

Fall 2020

Instructor: Dr. Bhagi Narahari



CSCI 2416 Fall 2020 Instruction team

- Instructor: Bhagi Narahari
- TA: William (Billy) Miller
  - 5-year BS+MS, BS CS Class of 2020
- Undergraduate TA (UTA): will be teaching a lab section
  - Sarah Morin – BS CS Class of 2021 (Senior)
  - Linnea Dierkshide – BS CS Class of 2022 (Junior)
- Learning Assistants (LA):
  - Graham Schock – BS CS Class of 2022 (Junior)
  - Catherine Meadows – BS CS Class of 2022 (Junior)

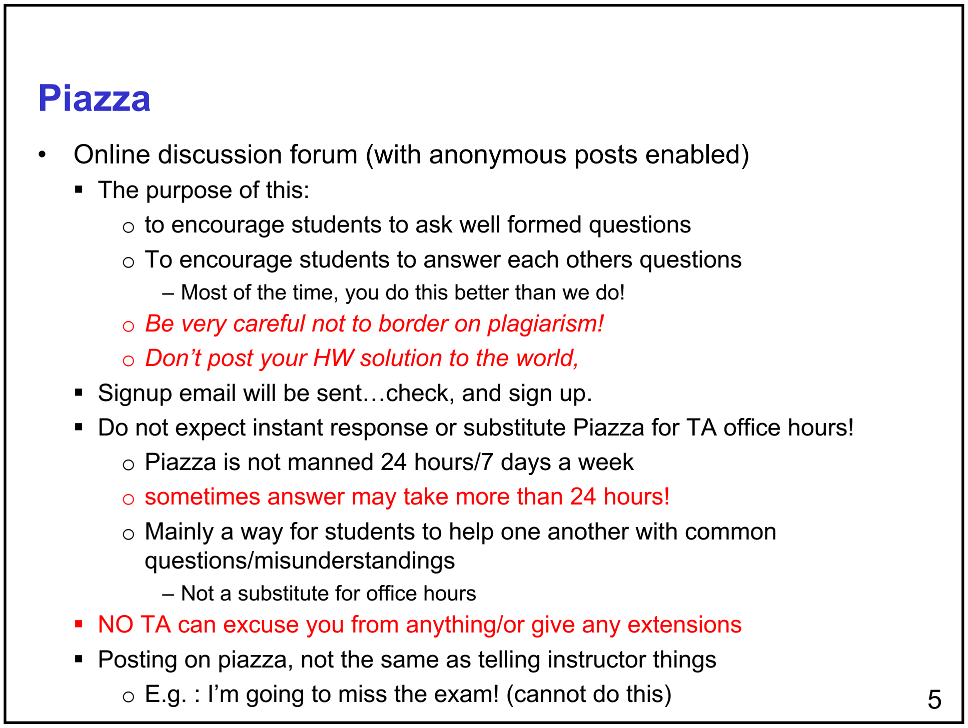
instruction team will have “office hours” and will be helping with in-class/in-lab activities

How is the course taught ?

- Lecture materials(Asynchronous) – Read/View **before** class
  - In some cases, videos of lecture topics
  - Slides, Notes, and examples (Circuits, Code,..)
- Synchronous lecture sessions
  - Blackboard collaborate
  - Summary of topics covered in videos & notes/slides
  - In-class discussions and exercises
    - Work on solving problems in breakout room with instruction team
- Lab sessions
  - Notes and Videos of lab tutorials posted – watch them before lab
  - Synchronous session during your lab section
    - Summary of topics
    - Exercises/Demos/Experiments
    - Will need to submit lab work for grading

Course Logistics….Where do I go to get course information & materials ?

- Blackboard:
  - Quizzes, Exams, Homeworks & Grades
  - Online class (lecture and lab) – recordings from these sessions
- Website: https://cs2461-2020.github.io/
  - Syllabus – schedule, grading criteria, contact info
    - Lecture notes(slides, exercises, code samples, circuits,…)
    - Videos linked from website
- Github: projects, and code submission
- Piazza: use as message board and announcements
  - Post questions – instructors or classmates can post answers
- Slack: direct messaging to instructors
  - when working on problems during class, default privacy (direct messaging)

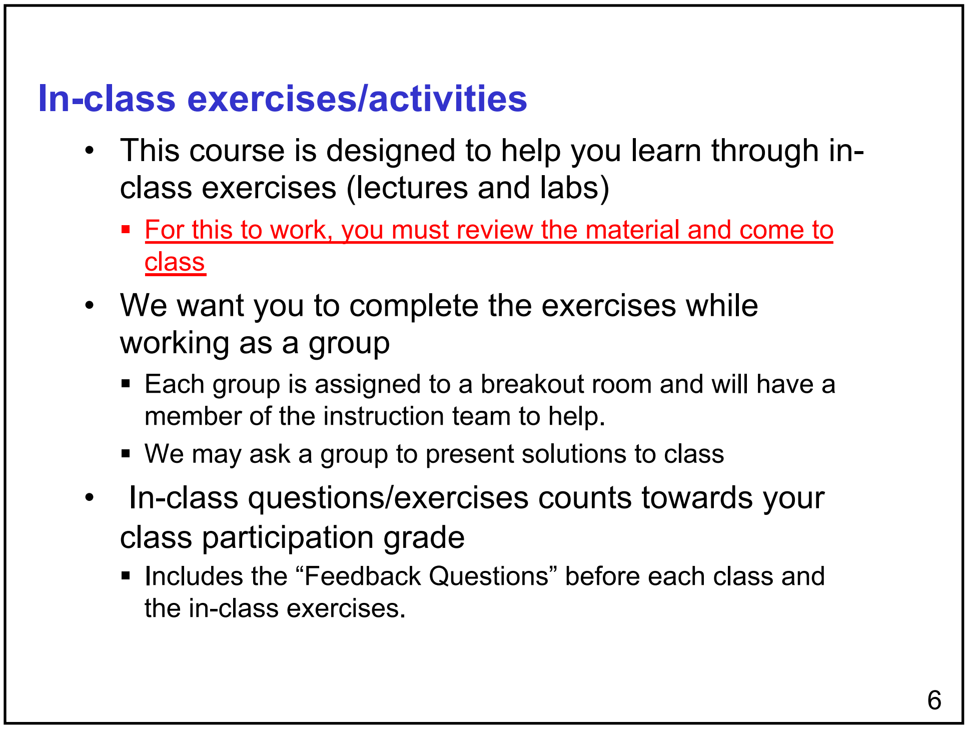


Piazza

- Online discussion forum (with anonymous posts enabled)
  - The purpose of this:
    - to encourage students to ask well formed questions
    - To encourage students to answer each others questions
      - Most of the time, you do this better than we do!
    - *Be very careful not to border on plagiarism!*
    - *Don’t post your HW solution to the world,*
  - Signup email will be sent…check, and sign up.
  - Do not expect instant response or substitute Piazza for TA office hours!
    - Piazza is not manned 24 hours/7 days a week
    - sometimes answer may take more than 24 hours!
    - Mainly a way for students to help one another with common questions/misunderstandings
      - Not a substitute for office hours
  - NO TA can excuse you from anything/or give any extensions
  - Posting on piazza, not the same as telling instructor things
    - E.g. : I’m going to miss the exam! (cannot do this)

5

5

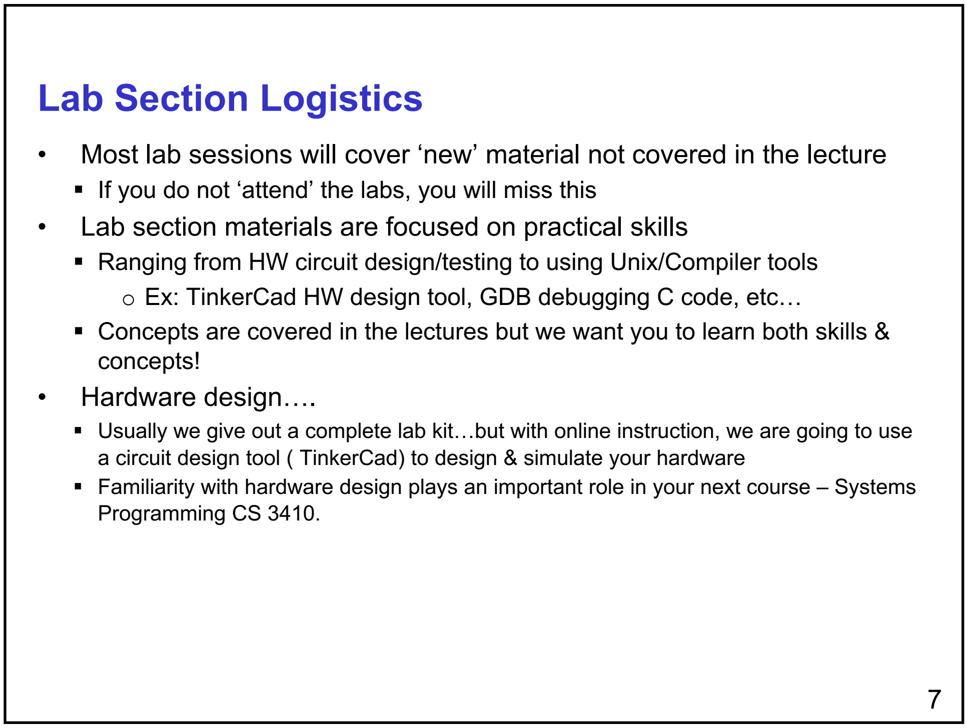


In-class exercises/activities

- This course is designed to help you learn through in-class exercises (lectures and labs)
  - For this to work, you must review the material and come to class
- We want you to complete the exercises while working as a group
  - Each group is assigned to a breakout room and will have a member of the instruction team to help.
  - We may ask a group to present solutions to class
- In-class questions/exercises counts towards your class participation grade
  - Includes the “Feedback Questions” before each class and the in-class exercises.

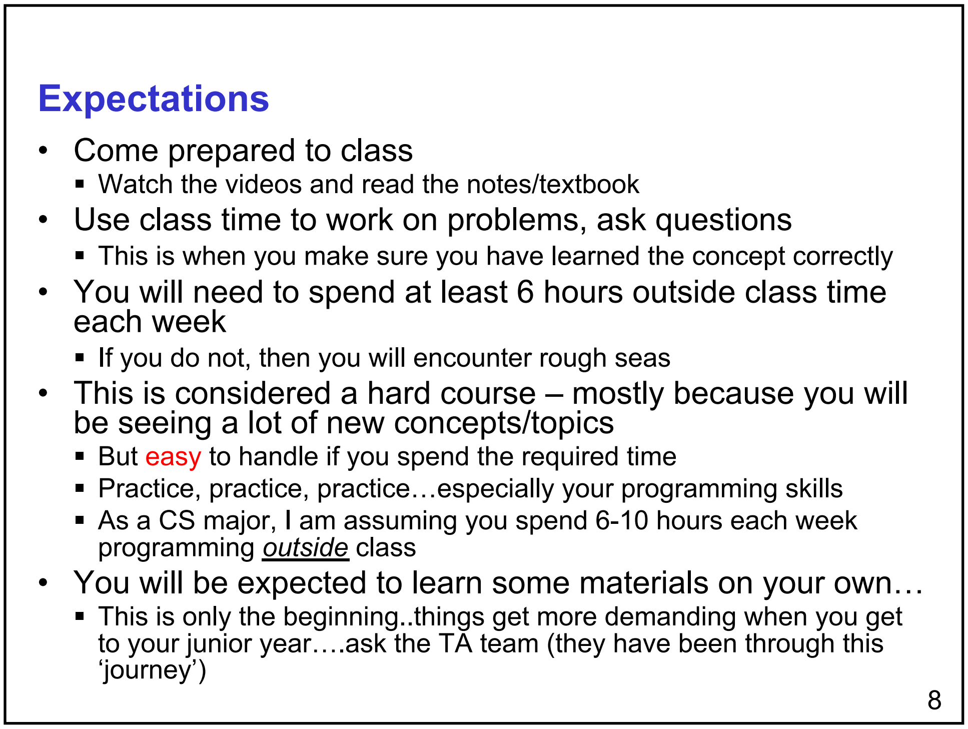
6

6



Lab Section Logistics

- Most lab sessions will cover ‘new’ material not covered in the lecture
  - If you do not ‘attend’ the labs, you will miss this
- Lab section materials are focused on practical skills
  - Ranging from HW circuit design/testing to using Unix/Compiler tools
    - Ex: TinkerCad HW design tool, GDB debugging C code, etc…
  - Concepts are covered in the lectures but we want you to learn both skills & concepts!
- Hardware design….
  - Usually we give out a complete lab kit…but with online instruction, we are going to use a circuit design tool (TinkerCad) to design & simulate your hardware
  - Familiarity with hardware design plays an important role in your next course – Systems Programming CS 3410.



Expectations

- Come prepared to class
  - Watch the videos and read the notes/textbook
- Use class time to work on problems, ask questions
  - This is when you make sure you have learned the concept correctly
- You will need to spend at least 6 hours outside class time each week
  - If you do not, then you will encounter rough seas
- This is considered a hard course – mostly because you will be seeing a lot of new concepts/topics
  - But easy to handle if you spend the required time
  - Practice, practice, practice…especially your programming skills
  - As a CS major, I am assuming you spend 6-10 hours each week programming ***outside*** class
- You will be expected to learn some materials on your own…
  - This is only the beginning..things get more demanding when you get to your junior year….ask the TA team (they have been through this ‘journey’)

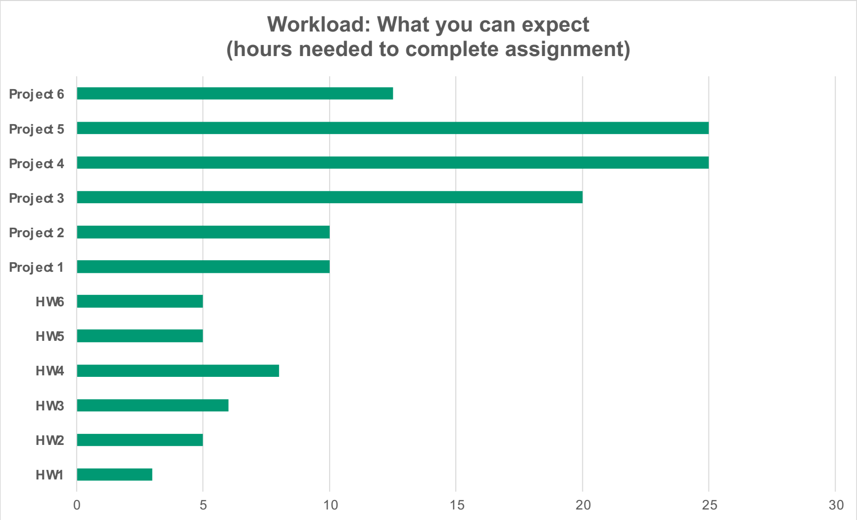
Textbooks/Software

- Intro. to Computing Systems, by Patt and Patel
  - Easy to read – fairly comprehensive.
- Other useful books: Unix for Programmers, C Programming
- LC3 simulator and C to LC3 compiler
  - Links on the course webpage (also on Textbook website)
- C compiler – gcc (need to use cygwin/Linux/Mac or SEASCF)
- Basic Unix Programming
- Hardware Simulators:
  - TinkerCad (for labs)
  - Logic simulators: CedarLogic (free S/W for Windows) or LogiSim(for Mac)

Course Schedule

- Part 1 (8-9 weeks) of the course spent on hardware stack and HW/SW Interface
  - From transistors to the design of a simple processor
    - Implementation of a simple processor ISA
  - Assembly programming
- Part 2 of the course (5-6 weeks) spent on C Prog Lang. and translation to Assembly
  - Quick review of C (you will cover some C in CS2113)
  - How are C constructs compiled into (LC3 assembly) machine language
  - Managing Memory
    - Stack
    - Heap
  - How to make your programs run faster

Course workload: What you’ve signed up for….

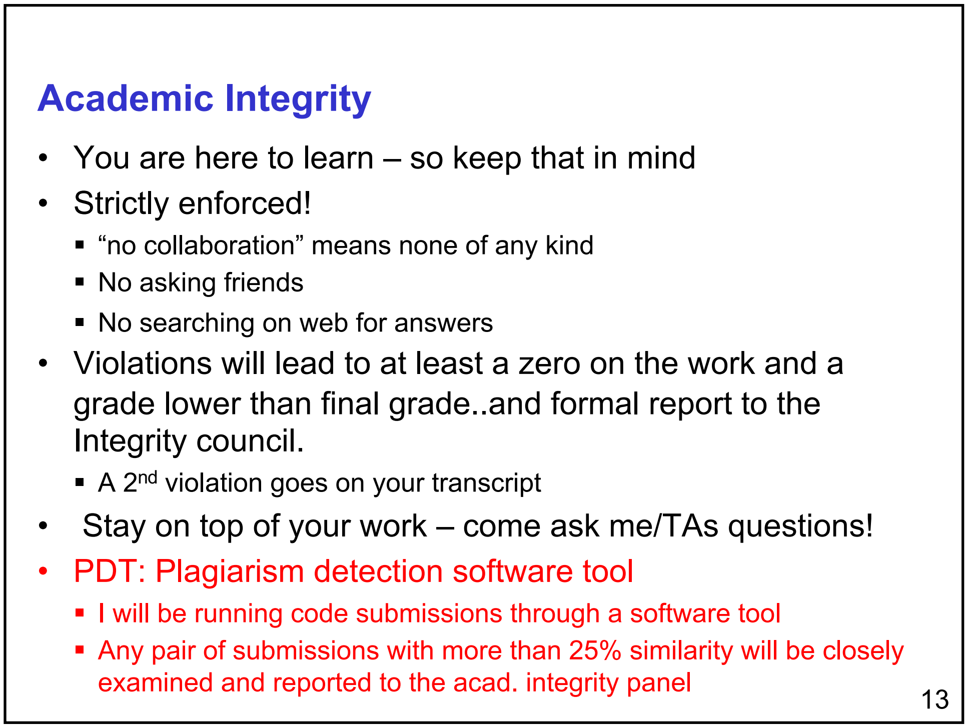


11

Requirements and Grading: Read website for details on how grade is computed

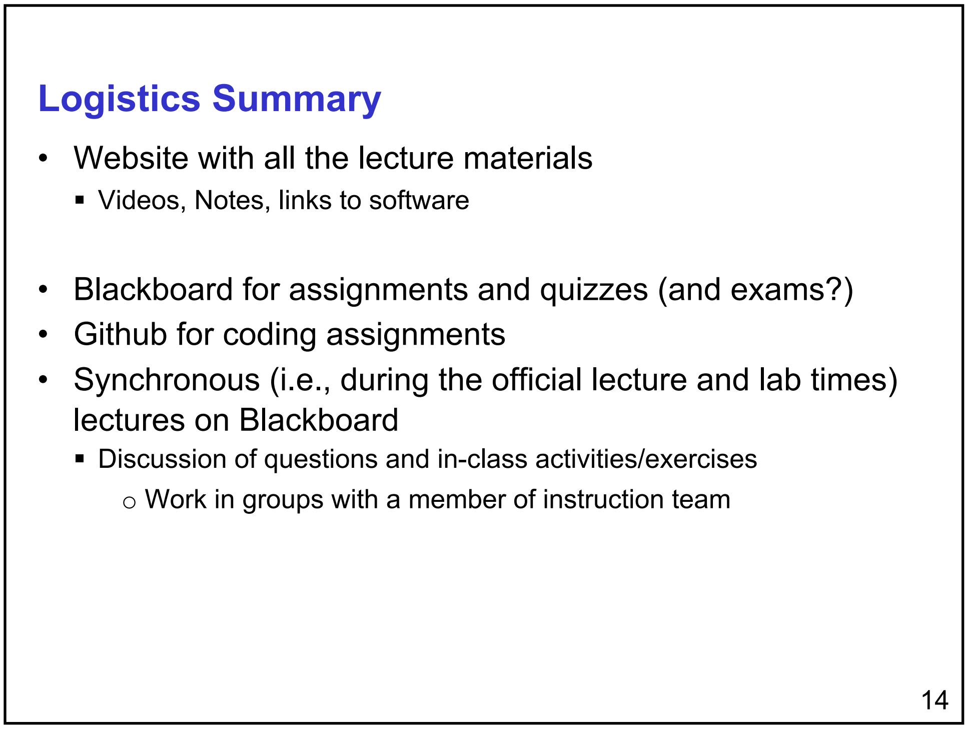
- 38% Exams: Two exams
  - Will be held approximately Weeks 7,12
  - Exam will also have an ‘interview’ (oral exam) component
    - Conducted by instructor and TA
- 20% Homework and Lab assignments
  - Lab assignments may require completion within lab time
    - No late submissions….except a “one time pass” of 36 hours
- 12% Class participation and Quizzes
  - 9 quizzes, will drop lowest score
    - **Start of class** – if you join late, you miss the quiz
  - Class participation – includes ‘Feedback Questions’ for each lecture and the inclass activities.
- 30% Projects and Team activities: about 6 projects
  - One of them will have a teamwork component.
  - You may be asked to demo and explain some of your projects – be prepared to be examined on any (random) aspect of your solution (code, design, etc.)

12



Academic Integrity

- You are here to learn – so keep that in mind
- Strictly enforced!
  - “no collaboration” means none of any kind
  - No asking friends
  - No searching on web for answers
- Violations will lead to at least a zero on the work and a grade lower than final grade..and formal report to the Integrity council.
  - A 2nd violation goes on your transcript
- Stay on top of your work – come ask me/TAs questions!
- PDT: Plagiarism detection software tool
  - I will be running code submissions through a software tool
  - Any pair of submissions with more than 25% similarity will be closely examined and reported to the acad. integrity panel



Logistics Summary

- Website with all the lecture materials
  - Videos, Notes, links to software
- Blackboard for assignments and quizzes (and exams?)
- Github for coding assignments
- Synchronous (i.e., during the official lecture and lab times) lectures on Blackboard
  - Discussion of questions and in-class activities/exercises
    - Work in groups with a member of instruction team