Welcome To CS15!
Welcome to CS 15 on Zoom!

- Zoom link is on course website
- We encourage you to download the PowerPoint slide deck before lecture – lets you see clearly and annotate
  - [http://cs.brown.edu/courses/cs015/](http://cs.brown.edu/courses/cs015/)

- We Zoom-record and give you web access to every lecture
  - for review
  - in case you have to miss a lecture
  - more about asynchronous lecture on next slide!
Accessibility for Virtual Learning

- **CS15 can be taken asynchronously**
  - register on C@B for asynchronous section – lecture will be watched asynchronously, section/lab and TA hours will still be synchronous
- **Section, Lab, and TA Hours offered at times that will accommodate all time zones**
- **Lecture questions will be answered on Piazza**
  - TAs will answer most questions via Piazza
  - Andy will ask for some questions live
    - use Zoom’s raise hand feature to ask a question
- **We are here for you!!**
  - reach out to TAs/Andy at any time – we want to help support you through this virtual semester
Our Hardware

• The Sunlab: 80 PCs running Debian Linux
  o not COVID-19 safe
• This year, all work will be done locally on your own computer
  o using GitHub to acquire and store code
  o IntelliJ IDEA – integrated development environment (IDE) we will use to write, compile, and run code
• Working From Home setup materials and instructional video released after lecture
  o come to TA hours for help setting up software!
CS15 is All That

- Teaches Object-Oriented Programming (OOP)
  - most common current programming methodology
  - Brown was earliest to switch to Java for intro courses almost two decades ago
  - still a dominant web programming language (e.g., Google’s Android)
- Teaches fundamental *problem-solving skills* useful in all disciplines
- Provides introduction to computer science concepts
- Is **intense**, but **fun**, especially with interactive graphics
- Uses games as a domain, but teaches skills applicable across domains
  - not a game design course (we have those too: 2D and 3D game engines)
Who is CS15 For?

- Students with varying levels of programming experience, including *NONE!*
  - however, CS15 still requires a *serious* commitment

- Most students have *little or no* programming experience, including the TAs and HTAs when they took the course!
  - let’s visualize this!

- Prospective CS concentrators, who will go on to CS16

- This is *not a weeding-out course*, but it is still time-consuming
  - don’t worry!! We expect lots of confusion in the beginning. All 35 TAs are here to help you through that initial confusion!
CS15 isn’t About Getting the Correct “Answer”

- It’s about the **process**, not just the final product!
  - design
    - planning efficient, effective designs for program structure
    - investing upfront (e.g., reviewing materials) saves time in long run
  - implementation
    - coding incrementally
  - debugging
    - diagnosing and fixing bugs/errors in code effectively
Diversity & Inclusion in CS15

- CS15 welcomes all, helps you succeed, and aims to build **community**. These additional groups are also here for you:

- Mosaic+ [mosaic.plus.brown@gmail.com](mailto:mosaic.plus.brown@gmail.com)
  - “created to foster Community, inspire Innovation, and provide opportunity to underrepresented minority students.”

- Women In Computer Science (WiCS) [wics@lists.cs.brown.edu](mailto:wics@lists.cs.brown.edu)
  - “formed by female undergraduate students at Brown in the late 1980s, The goal of WiCS is to increase the participation of women in the field of Computer Science.”

- Women in Science & Engineering (WiSE) [WiSE@brown.edu](mailto:WiSE@brown.edu)
  - “to encourage women who study in all science and engineering fields, by building a community of like-minded scholars that provides peer support on their journey to becoming successful scientists at Brown and beyond.”

- Our own CS15 mentorship program!
  - more on this later
Why Java

• Supports interactive OOP
• Syntax similar to C++ but simpler, cleaner, and more beginner-friendly
• Allows platform-independence: write once, run everywhere (in principle)
• One of the most prevalent languages in industry today, e.g., Android, web servers (others include C, C++, C#, Python, Ruby, etc.)
• Note: *not* the same as JavaScript, a less purely object-oriented language used commonly in web applications
• OOP is one of several programming paradigms – CS17 uses ReasonML and Racket for “functional programming”
Course Mechanics (1/4)

● **No** quizzes or exams!
  o no exam time pressure
  o no “grading on a curve”: you do the work, you get the grade you deserve!
    Thus A is by far the most common grade

● **8 Assignments**
  o programming assignments, some of which have a design component
  o from brief homework to Tetris and beyond!
  o choose from a selection of final projects, or create your own “indy” project
  o all programs must meet a baseline level of functionality to receive credit, lots of room for “bells and whistles” for fun and extra credit
  o all programs must be handed in with baseline functionality by end-of-semester!
Course Mechanics (2/4)

- Assignments are graded on a hand-in schedule
  - most assignments have early, on-time, and late hand-in
  - early hand-in: 2% increase to your grade
  - late hand-in: 8% decrease from your grade
  - all assignments must be handed in before the end of the course

- Google Form “clicker” questions during lecture
  - interactive questions to improve engagement and comprehension
  - accounts for 5% of your final grade
  - graded on completion for both synchronous and asynchronous students

- Weekly discussion/lab sections
  - graded on mini-assignments and participation
  - accounts for 12% of your final grade
Course Mechanics (3/4)

● Keys to success
  o increase in program size and complexity throughout the semester
  o you can’t procrastinate and then cram, unlike in some other courses
  o start early, start today, start yesterday!!!
  o other courses don’t teach you to tackle programs of this complexity

● TA Hours
  o 30 TAs and 5 Head TAs
  o 130+ TA hours of personalized help per week!!!
    ▪ more than in any other course!
    ▪ everyone struggles sooner or later, including the TAs – part of the learning process
    ▪ we strongly encourage you all to go to hours and get to know the TAs – it is integral to the course (and NOT a sign of weakness!)
Course Mechanics (4/4)

• CS15 thrives on your feedback

• Questions *highly* encouraged during lecture! And we will add Google Form “clicker” questions next week…

• We provide a lot of written material; YOU are responsible for digesting all of it
Major Changes This Year (1/2)

- New project – Leap Frog
  - more straightforward than the project it replaced
  - tested by all TAs!
  - hopefully sets students up to better understand the concepts it covers
Major Changes This Year (2/2)

- Pre-Lab Videos
  - review of relevant concepts for each lab
  - make sure students comprehend the material through brief quizzes
  - ensure students are prepared for lab
Alternatives to CS15 (1/3)

For Concentrators & Non-concentrators:

- CS17 (spring semester this year) – John Hughes
  - also, no prior experience required
  - multiple programming paradigms
  - multiple programming languages
    - Racket, ReasonML in CS17; then Java, Scala in CS18
  - mastery, not mystery → no magic
  - focus on problem-solving skills/strategies
    - emphasis on abstraction and scale
  - integrate programming with analysis of algorithms
  - multiple application areas (AI, databases, etc.)
  - pair programming for labs and projects
  - for more information on other CS courses:
    [http://cs.brown.edu/degrees/undergrad/whatcourse/](http://cs.brown.edu/degrees/undergrad/whatcourse/)
Summary of CS15/17 Choice

- Both will adequately prepare you for upper-level courses
- No prior experience needed for either, similar work loads
- Different material covered
  - CS15 – Object-Oriented Programming, CS17 – Functional Programming
  - CS15 is more practice-oriented, CS17 is more foundations-oriented
  - CS15 celebrates magic, while CS17 emphasizes no magic
    - CS15 has little reliance on TA support code, but uses JavaFX extensively
- CS15 allows collaboration on:
  - mini-assignments, sections, labs
- No pair programming on projects
  - but no tests
- CS15: games and skits
- Pick based on your taste and what appeals to you – you can’t go wrong!
Alternatives to CS15 (2/3)

For Concentrators & Non-concentrators:

- CS0111 (Fall) – Doug Woos
  - not offered this semester
  - no prior experience required
  - the first in a new introductory computing sequence that spreads the foundational concepts over three courses rather than two
    - “allow more time to combine CS with other studies & mastering the fundamentals”
  - functional programming and imperative programming
    - learn Pyret and Python
  - integrates programming with data science and discussion of use of digital information
  - less intensive workload
Alternatives to CS15 and CS17/19

For Non-concentrators:

- **CS20: The Digital World (Fall 2020) – Donald Stanford**
  - introduction to computing; little emphasis on programming
  - discusses computing topics such as artificial intelligence, IT security, and digital media
  - a small introduction to HTML, Photoshop, Access, and Python

- **CS30: Computation for the Humanities (Fall 2020) – Archita Agarwal**
  - introduce computational methods for problem solving in humanities/social sciences
  - investigate real world problems from the news, media, and research
  - topics include data-gathering, analysis, visualization, web-based interfaces, algorithms, scripting
Collaboration (1/5)

● Brown’s Academic Code

  "Academic achievement is evaluated on the basis of work that a student produces independently. A student who obtains credit for work, words, or ideas that are not the products of his or her own effort is dishonest and in violation of Brown’s Academic Code. Such dishonesty undermines the integrity of academic standards of the University. Infringement of the Academic Code entails penalties ranging from reprimand to suspension, dismissal, or expulsion from the University.”
Collaboration (2/5)

● CS15 Collaboration Guidelines
  o Lectures
    ▪ **always** allowed to review and discuss with your peers!
  o Mini-assignments
    ▪ collaboration and discussion are **allowed and indeed encouraged**!
  o Lab Section
    ▪ collaboration and discussion are **allowed and again encouraged**
  o Programming Assignments – “all written work must be your own”
    ▪ discussion allowed **only in Design Section and Conceptual Hours**!
Collaboration (3/5)

- MOSS (Measure of Software Similarity)
  - Stanford-hosted AI software used to detect plagiarism – it signals undue similarity and we hand-check the code
  - used across industries in multi-million dollar lawsuits to protect intellectual property
  - every year, MOSS finds multiple collaboration violations (we check multiple years!)
  - last year, half a dozen cases, all found guilty by UH’s official committee
  - punishments typically directed NC, parental notification
  - MOSS is very good at what it does – don’t even think of trying to outwit it! (which is more work than doing the assignment!)
  - we also check the web

If ever in doubt about what is allowed, ask a TA!
Better to NC an assignment or even the course than being accused (and likely convicted)!
Note: we have a Regret Policy
Collaboration (4/5)

The issue of collaboration in intro CS courses has been in the news in past years:

Possible cheating uncovered in popular Harvard computer class
By Travis Andersen and Brian MacQuarrie Globe Staff  May 05, 2017

As Computer Coding Classes Swell, So Does Cheating
By JESS BIDGOOD and  • MAY 29, 2017

Competitive environment drives culture of cheating in computer science classes
BY KATE HUANGPU | FEBRUARY 22, 2018, 3:43 AM
Collaboration (5/5)

- Collaboration is **not** worth the risk
  - start early and get help when you need it! Lots of resources available to help you succeed in this course