Welcome to CS15!

Welcome to Salomon 101!

- We encourage you to download the PowerPoint before lecture and bring your laptop – lets you see clearly and annotate
- We record and give you web access to every lecture
  - review
  - in case you have to miss a lecture
  - PowerPoint slides come with associated recording

Our Hardware

- The Lab: 80 high-end PCs
  - Intel Core i5 (3.5 GHz) CPUs with 8GB RAM
  - row 6: nVidia GeForce GTX 970 graphics cards
  - other machines: nVidia GeForce GTX 460 graphics cards
  - dual & wide screen Monitor Configurations

- File Servers
  - Terabytes of disk space for your programs

- And you can work from your dorm room on your own computer
  - This will be set up during first lab. Be sure to bring your laptop

CS15 is All That

- Teaches Object-Oriented Programming (OOP)
  - most common current programming methodology
  - Brown was earliest to switch to Java for intro courses more than a decade ago
  - AP courses teach Java as well
- Teaches fundamental problem solving skills useful in all disciplines
- Provides introduction to computer science concepts
- Is intense, but fun, especially with interactive graphics
- CS15 - Who is it for?
  - students with varying levels of programming experience, including NONE!
  - prospective CS concentrators, who will go on to CS16
  - anyone who wants an in-depth introduction to modern programming in a semester
Why Java
- Java
  - supports interactive OOP
  - syntax similar to C++ but simpler, cleaner, and more beginner-friendly
    - Java does a lot more bookkeeping and resource management for you
  - 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: it is not the same as JavaScript, a less purely object-oriented language used commonly in web applications

Alternatives to CS15 (2/3)
For Concentrators & Non-concentrators:
- CS19 (fall semester) - Shriram Krishnamurthi
  - starts off combined with CS17
  - must complete supplemental homework while enrolled in CS17
  - must do well in CS17 and CS19 supplements
  - will be invited into CS19 based on performance
  - enrollment in this course will occur approximately one month into the semester
  - uses Pyret
  - students still must learn Java before enrolling in CS32, either through self-study or CS18
  - For more information: http://cs.brown.edu/courses/cs019/2015/AFQ.html

Alternatives to CS15 (3/3)
For Concentrators & Non-concentrators:
- CS15/16, CS17/18, & CS19 fill concentration requirements.
- All qualify you to take upper level courses, but do cover different material, from different points of view
- Higher course number does not mean higher difficulty

Alternatives to CS15 (1/3)
For Concentrators & Non-concentrators:
- CS17 (fall semester) - John Hughes
  - Multiple programming paradigms
    - functional programming is unique to the 17/18 sequence
  - Multiple programming languages
    - Racket, OCaml in CS17; then Java, Scala in CS18
  - Racket as a first programming language
    - simple and clear syntax and semantics
  - Mastery, not mystery — no magic
  - Focus on problem-solving skills/strategies
  - Integrate programming with analysis of algorithms
  - Multiple application areas (AI, databases, etc.)
  - Pair programming
For more information on other CS courses: http://cs.brown.edu/courses/cs015/docs/Course.pdf
Alternatives to CS15 and CS17/19 (1/2)

For Non-concentrators:
- CS2: The Digital World (Fall semester) - Donald L. 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
- CS4: Introduction to Scientific Computing and Problem Solving (spring semester) - Daniel F. Potter
  - Use MatLab and some Python
  - Teaches techniques to solve scientific problems using computers

Alternatives to CS15 and CS17/19 (2/2)

For Non-concentrators:
- CS931: Introduction to Computation for the Humanities and Social Sciences (Fall and Spring semester) – Alexandra Papoutsaki, Jun Ki Lee
  - Specifically developed for humanities/social sciences concentrators
  - Investigate real-world problems from news, current research, books like Freakonomics
  - Covers data-gathering, analysis, algorithms, scripting, and more.

Course Mechanics

- **No quizzes or exams!**
  - No “grading on a curve”, thus A is by far the most common grade
  - No exam time pressure
- **10 Assignments**
  - 7 programming assignments, some of which have a design component
  - From 30-minute homeworks to Tetris and beyond!
  - Choose from a selection of final projects, or create your own indie project
  - All programs must meet a baseline level of functionality to receive credit, lots of room for “bells and whistles” for fun and extra credit

Course Mechanics (1/2)

- **Keys to success**
  - Start early, work steadily, don’t fall behind
  - You can’t cram, unlike most other courses
  - Exponential growth of program size throughout the semester
  - No other course teaches you to tackle programs of this size
- **TA Hours**
  - 36 TAs and 4 Head TAs
  - Over 150 TA hours of personalized help per week!!!
    - More than in any other course!
    - We strongly encourage you all to go to hours and get to know the TAs - it is integral to the course (and not an admission of failure!)
Course Mechanics (2/2)

- CS15 thrives on your feedback
- Questions *highly* encouraged during lecture!
- We provide a lot of written material; YOU are responsible for digesting all of it

Major Changes This Year (1/3)

- Switch to a more modern graphics library – JavaFX
- JavaFX has officially replaced SWING, which CS15 used up until last year
- Modern, clean graphics with lots of room for creativity

Major Changes This Year (2/3)

- More emphasis on debugging
  - we teach you how to find and fix errors on your own
  - learning to debug efficiently will save you lots of time!
- Revamped many of our labs

Major Changes This Year (3/3)

- Piazza – an additional resource for help
- A “curated forum” that allows you to submit private questions to TAs which, if of general interest, can later be published to everyone
Collaboration (1/3)

- 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 (2/3)

- CS15 philosophy on collaboration
  - Our experience is that beginning students learn best when solving problems on their own, with adequate help from experienced TAs
    - other intro courses (like CS 17 and 19) encourage collaboration;
    - advanced courses also have a more lenient policy
  - CS15 grade is entirely based on programs & homeworks, all your own work
  - zero tolerance for unacceptable forms of collaboration
  - collaboration policy and first homework clearly outline what's acceptable

Collaboration (3/3)

- 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
  - punishments typically directed NC, parental notification
    - last year, we prosecuted 13 cases, resulting in 11 convictions
  - in short: 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!