Eric Roberts
CS 106A

Handout #5
April 2, 2012

Introduction to CS 106A

Introduction to CS 106A

Eric Roberts
CS 106A
April 2, 2012

Why Study Computer Science

- The computing industry offers some of the best employment opportunities for college graduates in the United States today:
  - The number of jobs in the domestic software industry are at an all-time high and are projected to grow dramatically over the next decade.
  - Salaries for newly minted B.S. graduates in Computer Science are high, sometimes exceeding the $100,000 mark.
  - In 2005, *Money* magazine rated software engineer as the number one job in America.
  - Employment in this area is vital for the economy.
- Beyond its marketability, computer science is an intellectually challenging and highly creative discipline. It can also be a great deal of fun.
- In CS 106A, we will cover many of the ideas and skills you need to begin a career in this field. At the same time, we try as hard as we can not to lose sight of the creativity and fun, making sure you have ample opportunities for both.

CS 106A Staff

Professor: Eric Roberts
`eroberts@cs.stanford.edu`
Office Hours (Gates 202):
Tuesdays 9:30-11:30

Head TA: Jeremy Keeshin
`jkeeshin@cs.stanford.edu`
Office Hours (Gates 160):
TBA

Is CS 106A the Right Course?

CS 106A: Programming Methodology (ENGR 70A)

Introduction to the engineering of computer applications emphasizing modern software engineering principles: object-oriented design, decomposition, encapsulation, abstraction, and testing. Uses the Java programming language. Emphasis is on good programming style and the built-in facilities of the Java language. No prior programming experience required.

CS 106B: Programming Abstractions (ENGR 70B)

Abstraction and its relation to programming. Software engineering principles of data abstraction and modularity. Object-oriented programming, fundamental data structures (such as stacks, queues, sets) and data-directed design. Recursion and recursive data structures (linked lists, trees, graphs). Introduction to time and space complexity analysis. Uses the programming language C++ covering its basic facilities. Prerequisite: 106A or equivalent.

CS 106X: Programming Abstractions (Accelerated) (ENGR 70X)

Intensive version of 106B for students with a strong programming background interested in a rigorous treatment of the topics at an accelerated pace. Additional advanced material and more challenging projects. Prerequisite: excellence in 106A or equivalent, or consent of instructor.

Important Administrative Notes

- You ***must*** sign up for a section to be enrolled in the course. Section signups will start at 5:00P.M. on Thursday and close at 5:00P.M. on Saturday. Be sure to sign up during that time at

  `http://cs198.stanford.edu/section/`

- Undergraduates ***must*** take CS 106A for 5 units. Unfortunately, the default on Axess is 3 units, so make sure that you change this value as you register.
- All handouts, assignments, lecture slides, and announcements are posted on the course web site at

  `http://cs106a.stanford.edu/`

- Both the midterm and the final are given at two scheduled times as shown in Handout #2. Special arrangements can be made for those who cannot make either time.

Assignments in CS 106A

- Assignments in CS 106A are due at 5:00P.M. Assignments that come in after 5:00 will be considered late.
- Everyone in CS 106A starts the quarter with two "late days" that you can use at any time you need some extra time. In my courses, late days correspond to class meetings, so that, if an assignment is due on Wednesday and you turn it in on Friday, that counts as ***one*** late day.
- Extensions can be approved only by the TA, Chris Piech.
- Assignments are graded by your section leader, who discusses your work in an interactive, one-on-one grading session.
- Each assignment is given two grades: one on functionality and one on programming style. Style matters. Companies in Silicon Valley expect Stanford graduates to understand how to write code that other programmers can maintain.

The CS 106A Grading Scale

- Functionality and style grades for the assignments use the following scale:

✚✚	A submission so good it "makes you weep."
✚	Exceeds requirements.
✓✚	Satisfies all requirements of the assignment.
✓	Meets most requirements, but with some problems.
✓−	Some more serious problems.
−	Even worse than that.
−−	Why did you turn this in?

Contests

- CS 106A will have three contests as follows:
  - The Karel Contest associated with Assignment #1
  - The Graphics Contest associated with Assignment #3
  - The Adventure Contest associated with Assignment #6
- First prize in the contest is a score of 100% on one of the graded components of the course, typically the final exam.
- As an additional incentive, entering any of the contests gives you chances to win an additional grand prize in a random drawing at the end of the quarter.
- Entering a contest also earns "house points" for your class in the style of the Hogwarts School from Harry Potter.
- Securing a runner-up prize or an honorable mention on any contest gives you additional chances in the random drawing, as does having an assignment submitted as a ++ candidate.

Honor Code Rules

Rule 1: You must indicate on your submission any assistance you received.

Rule 2: You must not share actual program code with other students.

Rule 3: You must not look at solution sets or program code from other years

Rule 4: You must be prepared to explain any program code you submit.

Encouraging Academic Integrity

- No one likes exams. Unfortunately, as long as the rate of Honor Code violations remains high, we have no real option.
- This quarter, I will adopt the following strategy to encourage academic integrity. The weight of the final exam will be

  15% + 5% for each Honor Code case filed this quarter

  The weight assigned to the homework will be whatever is left after the announced weights are assigned to the various other components, subject to a minimum of 15%.
- For example, if no Honor Code cases come up this quarter, the final will count for 15% and the homework will count for 60%. If, however, there are three cases (as there were the last time I taught 106A), the final will count for 30% and the homework for 45%. And so on . . .

Meet Karel the Robot

- Karel the Robot was developed at Stanford by Richard Pattis.
- Karel's world is composed of streets and avenues numbered from the southwest corner. (As in Manhattan, streets run east-west and avenues run north-south.)
- Karel's world is surrounded by a solid wall through which it cannot move. Depending on the problem, there may also be walls in the interior of the world that block Karel's passage.
- The only other objects that exist in Karel's world are ***beepers***, which are small plastic cones that emit a quiet beeping noise.
- Initially, Karel understands only four primitive commands:

`move()`	Move forward one square
`turnLeft()`	Turn 90 degrees to the left
`pickBeeper()`	Pick up a beeper from the current square
`putBeeper()`	Put down a beeper on the current square

Your First Challenge

- How would you program Karel to pick up the beeper and transport it to the top of the ledge? Karel should drop the beeper at the corner of 2nd Street and 4th Avenue and then continue one more corner to the east, ending up on 5th Avenue.