Overview

- Goals of This Course
- Interactions
  - Lectures, Recitations, Office Hours
- Assessment
  - Quizzes, Homeworks, Exams
- A Mysterious Function
Goals

Computational Thinking

Programming  Algorithms
Computational Thinking

- Specification vs. implementation; correctness
- Logical vs. operational reasoning
- Abstraction and interfaces
- Loop and data structure invariants
- Reasoning about resource bounds
Programming Skills

- Transformation of algorithmic ideas into correct imperative code
- Specify, write, test, debug, (re)factor code in the small
- Some familiarity with Unix tools and C
Programming Language

- CO: a small safe subset* of C
  - int, bool, char, string, arrays, pointers, structs
- Essential algorithmic and programming ideas
- Relatively close to machine (imperative)
- Sound reasoning with contracts
- Transition to C near end of course
Algorithmic Ideas

- Asymptotic complexity
  - time/space/parallel
  - worst case/average case
  - important classes: $O(1)$, $O(\log n)$, $O(n \log n)$, $O(n^k)$, $O(2^n)$

- Divide-and-conquer

- P vs NP [recently in the news!]

- Emphasis on imperative prog’$s$, ephemeral data struct’$s$
Concrete Algorithms
(sample, not a promise)

- Basic arithmetic
- Binary search, sorting
- Stacks and queues, priority queues
- Binary trees, dictionaries, maps, sets, tries
- Hashing, hash tables
- Graphs, reachability, shortest path, spanning trees
- Satisfiability (SAT)
Role in Curriculum

- 15-150 Principles of Functional Programming
- 15-213 Introduction to Computer Systems
- 15-210 Fundamental Alg’s & Data Struct’s
- 15-214 Principles of Software Systems
Overview

- Goals of this course
- Interactions
  - Lectures, Recitations, Office Hours
- Assessment
  - Quizzes, Homeworks, Exams
- A Mysterious Function
Lectures

- Please be here, please be active
- Ask and answer questions, pay attention
- No textbook, new course, ...
- Laptops for note-taking only
- No surfing, email, games, ...
- Too distracting for everyone else
Recitations

- Reinforce lecture material
- Problem solving
- How-to programming and tool support
- Get to know your instructor
Office Hours

- We like to see you!
- Any questions and issues with course
- See web page for current hours and location
- Cluster help available Tue & Thu 6:30–9:30!
On-line Communication

- Blackboard for grades, quizzes, email announcements
- Bboard cyrus.academic.cs.15-122
- Email to me, TA, or CA
- Cluster Linux machines for and /afs for assignments
Overview

- Goals of this course
- Interactions
  - Lectures, Recitations, Office Hours
- Assessment
  - Quizzes, Homeworks, Exams
- A Mysterious Function
Quizzes

- Test basic understanding
- On-line on Blackboard, auto-graded
- Due Monday night(!)
- 8 quizzes, drop lowest score
- Total of $7 \times 15 \approx 100$ pts
Midterms

- Test functional understanding of material
- During lecture period (80 mins)
- Closed book, closed laptop, 1 sheet of notes
- Total of 2 * 100 = 200 pts
Final

- Testing cumulative mastery of material
- Three hours during final exam period
- Closed book, closed laptop, 1 sheet of notes
- Total of 250 points
Assignments

- Weekly assignment (out Thu, due Thu)
- Apply material in problem solving context
- Combination of written and programming
- Hand-in start of lecture (written) & online (prog.)
- Total of 3 late days on prog, none on written
  - Max of 1 late day per assignment
- Total of 7 * 50 + 1 * 100 = 450 pts
Academic Integrity

- Quizzes, exams, homework must be your own.
- OK: discussion of course material, practice problems, study sessions.
- Not OK: copying or discussing answers, looking at or copying each others code (even parts).
- Tomorrow in recitation: read and sign academic integrity policy for this class; ask when in doubt.
- University policy will be applied rigorously!
Overview

- Goals of this course
- Interactions
  - Lectures, Recitations, Office Hours
- Assessment
  - Quizzes, Homeworks, Exams
- A Mysterious Function
Bug Report!

```c
int f (int n) {
    int i = 0; int k = 0;
    while (k <= n) {
        k += (i<<1) + 1;
        i++;
    }
    return i-1;
}
```