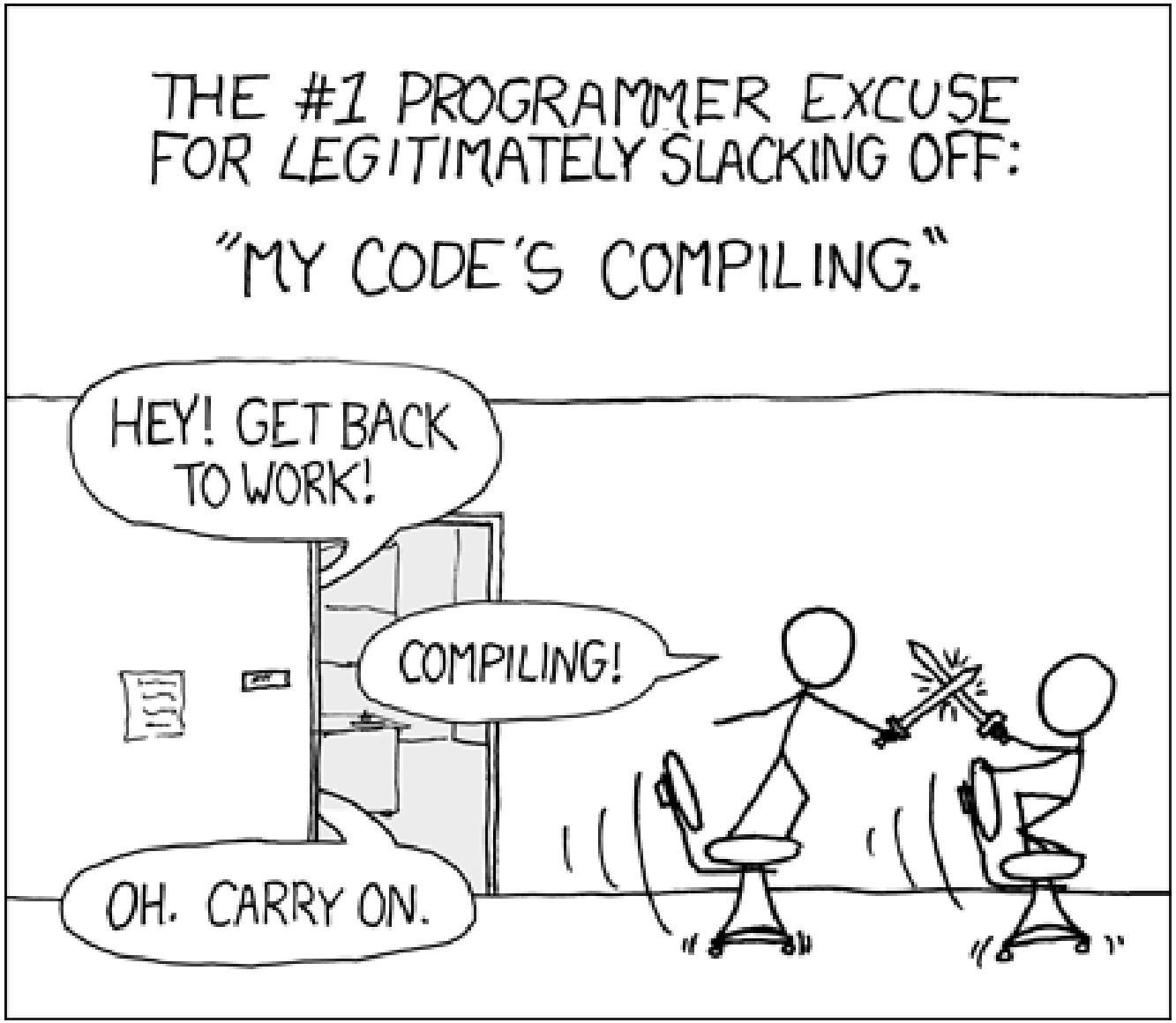


http://xkcd.com/303/

CSE 504: Compiler Design

http://www.cs.stonybrook.edu/~cse504/

Mon., Wed. 2:30pm – 3:50pm

Harriman Hall 116

C. R. Ramakrishnan

e-mail: cram@cs.stonybrook.edu

Office: 1420, Computer Science

Office Hours: Tue. 12noon – 1pm; Wed. 11:30am – 1pm; or by appointment

http://www.cs.stonybrook.edu/~cram

Last modified: Mon Jan 26 2015 at 17:11:54 EST
Version: 1.4 08:02:42 2013/02/05
Compiled at 17:16 on 2015/01/26

Prerequisites

- Courses:
  - **CSE 219** or **260**: Advanced Programming in an HLL
  - **CSE 220:** Systems-Level Programming
  - **CSE 303:** Automata Theory
  - At least one of the following:
    - **CSE 304:** Compiler Design (undergraduate)
    - **CSE 307:** Principles of Programming Languages
- Programming Experience:
  - Significant programming experience in multiple languages.
  - Ability (and a willingness) to adapt to new languages and tool chains.

Course Outcomes

At the end of the course, students are expected to have:

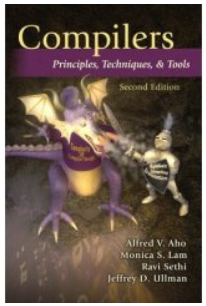
- An ability to use of formal attributed grammars for specifying the syntax and semantics of programming languages.
- Working knowledge of the major phases of compilation, particularly lexical analysis, parsing, semantic analysis, and code generation.
- An ability to design and implement a significant portion of a compiler for a language chosen by the instructor.

Informal Statement of Course Objective: To learn the process of translating a modern high-level language to executable code.

- Learn the fundamental techniques from lectures, text book and exercises from the book.
- Apply these techniques in practice to construct *a fully working compiler* for a non-trivial object-oriented language called `Proto`.

Organization

- Concepts and basic ideas in the lectures.



Aho, Lam, Sethi, Ullman
Compilers: Principles, Techniques, and Tools
(2nd ed., Addison Wesley)

- Concrete implementation in a series of programming homework assignments.

  Unlike earlier years (and similar courses in many other places)
  ... *we'll build a series of working compilers*
  ... progressively adding language and compiler features.

Grading

Homework assignments	35%
Informal project presentation	10%
One mid-term exam	25%
Comprehensive final exam	30%

Letter grades:

- 80% or above: definite A, provided:
  - at least 5 non-trivial homework submissions
  - 70% or better in the exams
- 50% or above: definite C+ or better provided:
  - at least 3 non-trivial homework submissions
  - 35% or better in the exams

The Rules of the Game

Programming assignments:

- Individual work.
- Late submissions (submitted up to 3 days late) will be graded normally, but have the following maximum score limit:

Late Days	**Max.**
On time	100%
1 day late	75%
2 days late	50%
3 days late	25%
$\geq$ 4 days late	0%

- No late penalty (up to 3 days) for 1 assignment of your choice! You should send me mail ahead of the submission deadline if you want to use this penalty-free late days for any assignment.

Your Responsibilities

- Limit discussion of homework assignments to *problems*, not *solutions*.
- Cheating, illegal collaboration and plagiarism will be treated with maximum seriousness.
  - Grad students will be referred to the Graduate Program Director.

  Conviction of academic dishonesty carries substantial penalties, such as receiving an 'F' grade, and/or expulsion from the University. See syllabus on course web page for details.

The Importance of Being Earnest

The Importance of Being Earnest

U.S. NEWS

DISCUSS AS:

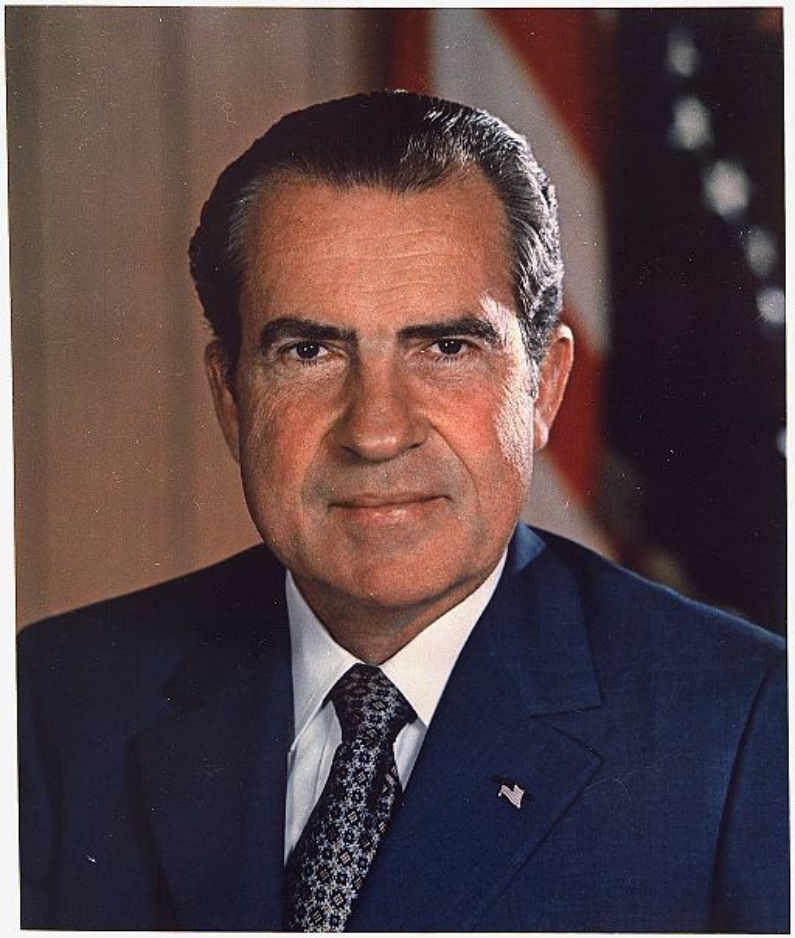
Sign in

92 nuclear missile officers implicated in cheating scandal, Air Force says

Thursday Jan 30, 2014 12:25 PM

http://lmgtfy.com/?q=nuclear+missile+officers+cheating

The Importance of Being Earnest



http://commons.wikimedia.org/wiki/File%3ARichard_M._Nixon%2C_ca._1935_-_1982_-_NARA_-_530679.jpg

The Importance of Being Earnest

http://commons.wikimedia.org/wiki/File%3ALance_Armstrong_Aviano.jpg

The Importance of Being Earnest

The New York Times

PRO FOOTBALL

League Probing Patriots' Possible Use of Deflated Balls

By REUTERS
JAN. 19, 2015, 11:20 A.M. E.S.T.

http://www.nytimes.com/reuters/2015/01/19/sports/football/19reuters-nfl-patriots.html

Effort Level

The effort required by this course is **High**

Effort Level

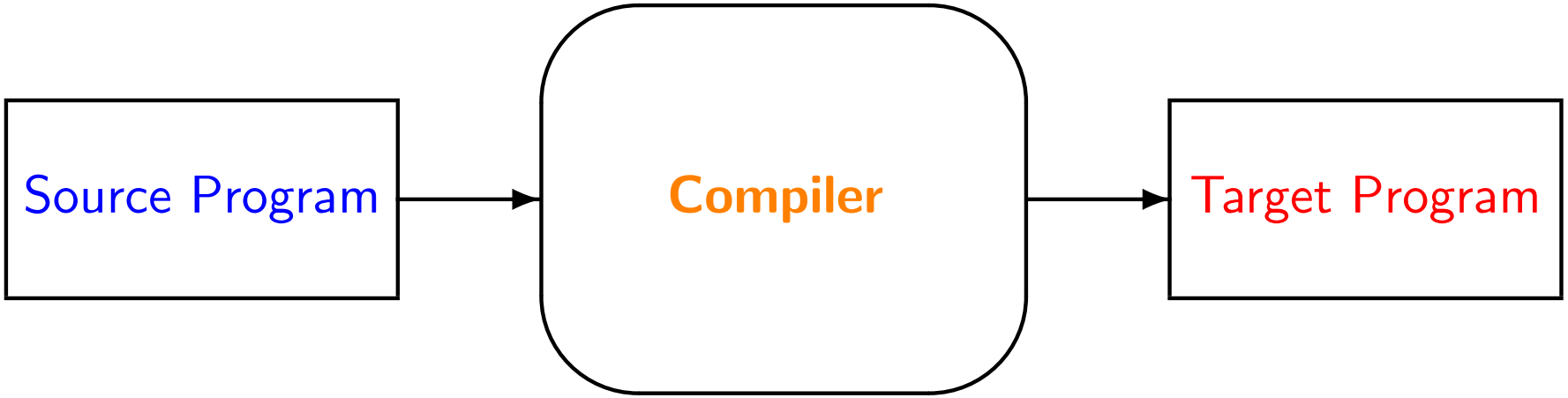
The effort required by this course is **High**

but so are the rewards:

- Hands on experience in large-scale programming (this semester: in Python???).
- Exposure to inner workings of Object-Oriented and Functional programming.
- In-depth knowledge of *how* programs written in high-level languages are translated and executed.

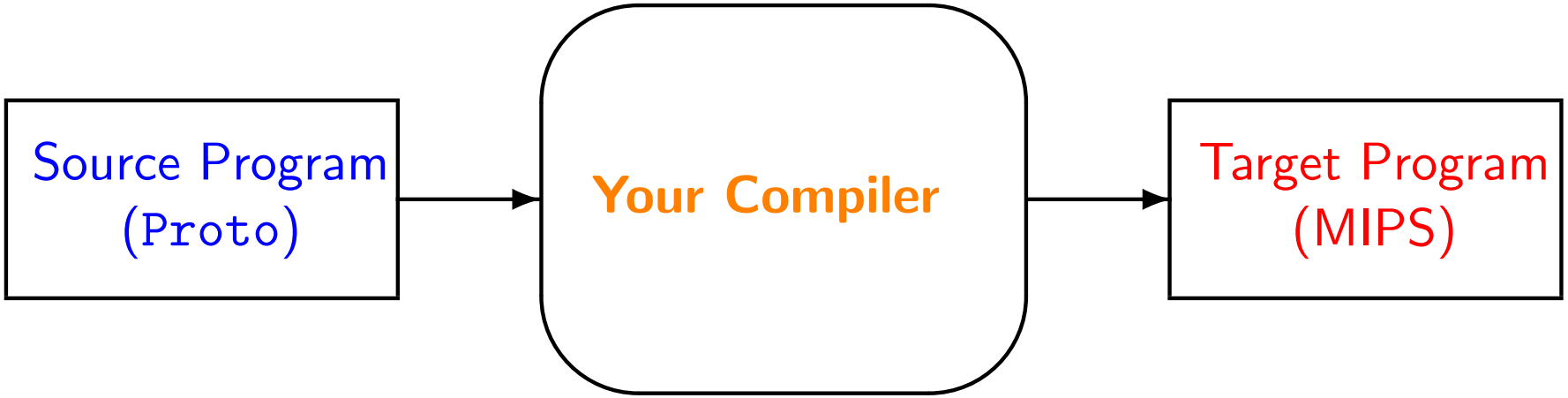
Course Structure

- Compilers in this course will be written in **???** (language to be decided).
- You will write compilers for a sequence of increasingly complex languages
  - all of which will be confusingly called as `Proto`.
- Your compilers will translate `Proto` programs to MIPS code
  - and run the result using the MIPS simulator SPIM.



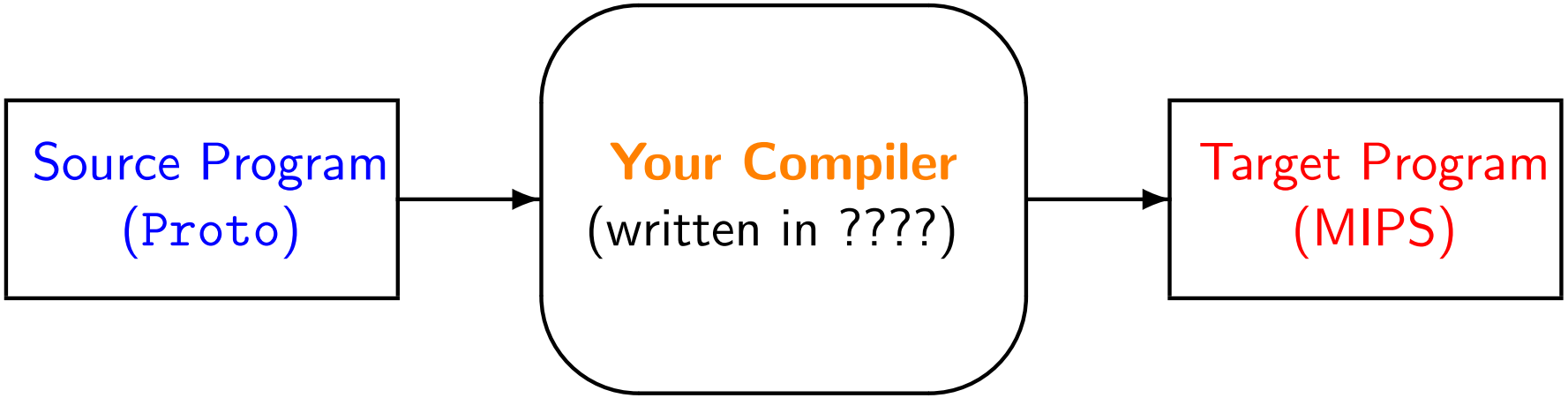
Course Structure

- Compilers in this course will be written in **???** (language to be decided).
- You will write compilers for a sequence of increasingly complex languages
  - all of which will be confusingly called as `Proto`.
- Your compilers will translate `Proto` programs to MIPS code
  - and run the result using the MIPS simulator SPIM.



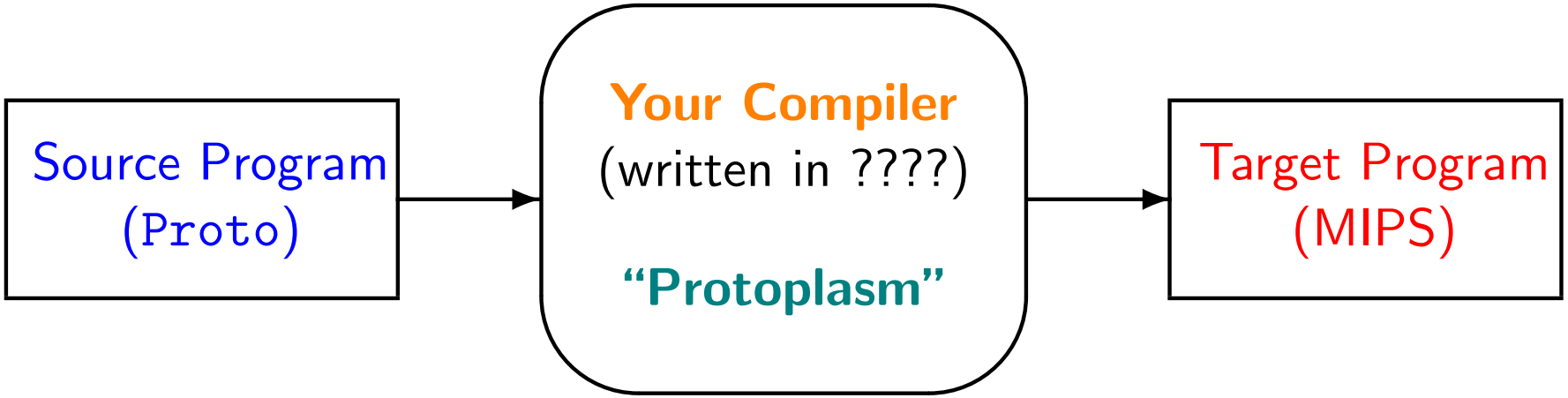
Course Structure

- Compilers in this course will be written in **???** (language to be decided).
- You will write compilers for a sequence of increasingly complex languages
  - all of which will be confusingly called as `Proto`.
- Your compilers will translate `Proto` programs to MIPS code
  - and run the result using the MIPS simulator SPIM.



Course Structure

- Compilers in this course will be written in **???** (language to be decided).
- You will write compilers for a sequence of increasingly complex languages
  - all of which will be confusingly called as `Proto`.
- Your compilers will translate `Proto` programs to MIPS code
  - and run the result using the MIPS simulator SPIM.



Proto

This is a tentative plan for the language `Proto`. Each language version adds features to the earlier ones.

1. **Proto** `Proto`, a calculator language: Integer constants, variables over integers, integer arithmetic expressions, sequences of assignments.
2. Control structures: Conditional statements, loops.
3. Data structures: Other basic types such as Booleans; structured types such as records and arrays.
4. Procedures: Functions and procedures, variable and type declarations.
5. Objects: Inheritance and overloading.

Compiler for each language level will be a separate homework assignment.

MIPS

- A RISC machine that is very widely used in Assembly Language Programming and Architecture courses (including our own CSE 220).
- SPIM is a MIPS simulator, also widely used for running and debugging MIPS assembly programs.
- I'll assume familiarity with programming in assembly language in general, and MIPS in particular.
  - If you are have not familiar, be prepared to pick it up really quickly.

Course Software

- SPIM is available for most machines; download and install it on your machine for testing the results of your compiler.
- Get `git` if you don't have it already.

The course web site has links to tutorials and documentation for MIPS assembly language, SPIM and `git`.

If you find more useful references, let me know!

Instructional Support

All course material, including homework handouts, notes, etc. will be posted on the course web site.

We will use the Blackboard system for

- **Course Announcements** (check these regularly)
- **Course Discussion Board**
- **Feedback and Grades** on assignments

I'm going to experiment with submitting assignments via `git`.

- If that works out for the first assignment, we'll use it for the remainder of the course.
- If there are issues with it, we'll default to the Blackboard for assignment submission.

Course Discussion Board

- Use this to discuss any course-related material: lectures, homework problems, exams, etc.
- If you have any questions on the material, first check to see if any one else had the same question as you have, and whether the question has been answered already; otherwise post the question on the board.
- We'll try to answer all questions on the board asap: within 24 hrs for normal days, and much quicker near exam/homework deadlines.

Questions

How to contact course staff:

- Staff contact information is on the web page/Blackboard.
- Post your question on the discussion board.
- Meet me during my office hours (or fix an appointment).
- Send me email. (Post on discussion board unless the question is personal.)
- Contact TA (to be announced) by visiting him/her during office hours or by email.
- Grading related questions: send email to the TA (who grades your homework assignments) first; if unresolved, send me email.