### Course Goals

This course is intended to make you a better software programmer by providing an introduction to the processes and considerations of Software Engineering. The following are some of the primary topics:

#### Software Development Practices:
- Know the values, principles, and practices of the Agile design process
- Experience the benefits of Paired Programming
- Understand Test-Driven Development (TDD)
- Be able to identify "rot" in code, and know how to refactor to improve
- Learn event-driven programming in Java
- Be exposed to design principles and patterns

#### Software Engineering Process Tools:
- Use version control (git) to support team programming projects
- Use UML (uniform modeling language) to explore and communicate design decisions

#### Textbooks

The following books are recommended.


Additional Recommended Works:
The following books contain additional information that you may find useful:


Student Evaluation
Course grades will be based on the following:

- Labs, exercises, projects: 50%
- Class participation: 5%
- Exams: 45%

Course Format
The format of this class will consist of reading assignments, in-class exercises, coding assignments, and exams. Software development is generally a group effort, and this class will encourage discussion and collaboration among all class attendees. The schedule will be flexible, and additional topics will be introduced at various points in the semester.

Attendance Policy
Since most of the work for this class will be done in pairs or teams, class attendance is very important, especially on days when we are completing assignments that we have already begun. If you don’t show up then your partner(s) must complete the assignment on their own. To encourage participation and attendance, I will pass around an attendance sheet each day. In order to get full participation credit, you must also post on piazza.

Homework Policy
Late work is _strongly_ discouraged. Late assignments will be penalized 10% per day. For small assignments (less than 10 points), the late penalty is 1 point per day. _No_ assignments will be accepted more than 2 days late.

Incorrect assignments (e.g., submitting the wrong file, forgetting to include your partner's name, etc.) will be assessed a penalty of 10%. Programs that do not compile/run will receive a deduction of 10-20% (in addition to points lost for missing features).

Computer Facilities
Eclipse is installed on most (maybe all) computer facilities on campus, and can also be downloaded onto your own personal computers.
Information Sharing Policy

All turned-in assignments and tests must be your own work. Although group assignments will emphasize learning to program in groups, each student is expected to contribute individually, and uphold both the Honor Code and the Collaboration Policy for Programming Projects, below:

Honor Code

I pledge to uphold the high standards of academic ethics and integrity expressed by the of Mines Student Honor Code by which I am bound. In particular, 'I will not misrepresent the work of others as my own, nor will I give or receive unauthorized assistance in the performance of academic coursework.' I understand that my instructor will report any infraction of academic integrity to the Department Head and that any such matter will be investigated and prosecuted fully.

Collaboration Policy for Programming Projects in CS Courses

The following policy exists for all CS courses in the EECS department. This policy is a minimum standard; your instructor may decide to augment this policy.

1. If the project is an individual effort project, you are not allowed to give code you have developed to another student or use code provided by another student. If the project is a group project, you are only allowed to share code with your group members.

2. You are encouraged to discuss programming projects with other students in the class, as long as the following rules are followed:
   a) You view another student's code only for the purpose of offering/receiving debugging assistance. Students can only give advice on what problems to look for; they cannot debug your code for you. All changes to your code must be made by you.
   b) Your discussion is subject to the empty hands policy, which means you leave the discussion without any record [electronic, mechanical or otherwise] of the discussion.

3. To prevent unintended sharing, any code stored in a hosted repository (e.g., on github) must be private. For group projects, your team members may, of course, be collaborators.

4. Any material from any outside source such as books, projects, and in particular, from the Web, should be properly referenced and should only be used if specifically allowed for the assignment.

5. If you are aware of students violating this policy, you are encouraged to inform the professor of the course. Violating this policy will be treated as an academic misconduct for all students involved. See the Student Handbook for details on academic dishonesty.