CS 30700: Software Engineering
Spring 2017

**Class:**
MWF 11:30 AM, Room ME 1061

**Course Web Page:**
http://courses.cs.purdue.edu/cs30700:spring17:start

**Course Newsgroup:**
https://piazza.com/purdue/spring2017/cs30700

**Instructor:**
Prof. Jeff Turkstra, jeff@purdue.edu, HAAS 128, 49-63088.

**Office Hours:**
MWF 12:30pm-1:30pm, TTh 3:00pm-4:20pm

**Project Coordinators:**
This course has nine project coordinators. The names, email addresses, and offices for them are given below. All TA office hours are by appointment.

<table>
<thead>
<tr>
<th>Name</th>
<th>Email</th>
<th>Office</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keehwan Park</td>
<td><a href="mailto:park451@purdue.edu">park451@purdue.edu</a></td>
<td>HAAS G50</td>
</tr>
<tr>
<td>Head TA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hasini Gunasinghe</td>
<td><a href="mailto:huralali@purdue.edu">huralali@purdue.edu</a></td>
<td>HAAS G50</td>
</tr>
<tr>
<td>Asmaa Sallam</td>
<td><a href="mailto:asallam@purdue.edu">asallam@purdue.edu</a></td>
<td>HAAS G50</td>
</tr>
<tr>
<td>Anoop Ranganathan Santhosh</td>
<td><a href="mailto:asanthos@purdue.edu">asanthos@purdue.edu</a></td>
<td>HAAS G50</td>
</tr>
<tr>
<td>Garrick Buckley</td>
<td><a href="mailto:gbuckle@purdue.edu">gbuckle@purdue.edu</a></td>
<td>HAAS G50</td>
</tr>
<tr>
<td>Dennis Chia</td>
<td><a href="mailto:dchia@purdue.edu">dchia@purdue.edu</a></td>
<td>HAAS G50</td>
</tr>
<tr>
<td>Aviral Mansingka</td>
<td><a href="mailto:amansin@purdue.edu">amansin@purdue.edu</a></td>
<td>HAAS G50</td>
</tr>
<tr>
<td>Matthew Page</td>
<td><a href="mailto:page38@purdue.edu">page38@purdue.edu</a></td>
<td>HAAS G50</td>
</tr>
<tr>
<td>Sahil Pujari</td>
<td><a href="mailto:pujari@purdue.edu">pujari@purdue.edu</a></td>
<td>HAAS G50</td>
</tr>
<tr>
<td>Nihanshu Purohit</td>
<td><a href="mailto:npurohit@purdue.edu">npurohit@purdue.edu</a></td>
<td>HAAS G50</td>
</tr>
<tr>
<td>Tori Shurman</td>
<td><a href="mailto:vshurman@purdue.edu">vshurman@purdue.edu</a></td>
<td>HAAS G50</td>
</tr>
</tbody>
</table>
Text:
Optional
Any up-to-date Software Engineering book (borrowed from a friend or from a library or in electronic format) will be fine. Make sure the book covers the Agile software development process – specifically, scrum.

Essentials of Software Engineering; Tsui, Karam, and Bernal; Jones & Bartlett, 2014

Beginning Software Engineering; Rod Stephens; John Wiley & Sons, 2015

Prerequisites:
Data Structures and Algorithms – CS 25100
Foundations of Computer Science – CS 18200
Programming in C – CS 24000
Problem Solving and Object-Oriented Programming – CS 18000
Programming proficiency is absolutely required

Course Outcomes:
A student who successfully fulfills the course requirements will:
1. understand the software development process
2. understand the advantages and disadvantages of current software life cycle models
3. use current tools and methods to plan analyze, design, test, measure, and manage software projects
4. learn about and use version control systems
5. understand that good people are one of, if not the most important, requirements for a successful project
6. learn how to work on a team project

Class Attendance
You are expected to attend all classes. Attendance will be recorded for randomly selected class sessions. If you choose to attend class, please arrive in the classroom on time. You are expected to be quiet in class. If you must miss a class, you are responsible for procuring any material, information, handouts, announcements, etc., that you missed.

Preparation for Lectures
You should try to read over the relevant pages in the course text before arrival. Additionally, you are expected to check your email and the course website regularly. Here is the tentative lecture schedule and corresponding chapters in the course text:
**Homework**
Homework assignments are assigned as needed to support class material and are due on the date given in the assignment. These are written assignments that are submitted electronically.

**Quizzes**
There will be a number of unannounced, 5 to 10 minute quizzes in class. The lowest grade will be dropped at the end of the semester. A score of zero will be given in case of absence.

**Project**
The goal of the project component of this course is to establish a team of four to six people and engage in an intense software engineering effort following the scrum methodology. A project charter followed by a requirements document will be generated. Subsequently, there will be three “sprints,” each preceded by a planning document and proceeded by a retrospective. A review will take place during each sprint where the project coordinator will assess the efforts and accomplishments of the team. There will be a final project presentation at the conclusion of the semester. Documents submitted beyond established deadlines will incur a 10% penalty per day.

**Examination**
The exam will be closed book and closed notes. You must solve the exam problems yourself, without any help (knowing or unknowing) from any other student. You must not seek any knowledge in advance of the test questions (beyond that given in class) and must report any advance knowledge of the test questions by any student that you are aware of. You must not allow any other student access to your solutions during the exam. If the seating situation makes this difficult, please inform the instructor or TAs.
Regrades
Problems regarding grading of assignments and the exam must be resolved within one week after the graded work has been returned to you. It is your responsibility to pick up the graded work on time. Grades will not be modified after the one week period.

Make-up Examination Policy
Make-up exams will be given only in the most extreme circumstances and require certification for such circumstances. Eg, a medical doctor's statement certifying that the student is unable to attend the scheduled exam. Any travel (including interview trips), load from work or from other classes, failed alarm clocks, or simply not being able to make it to the exam will not be grounds for a make-up. If you have any recurring medical problems that may unexpectedly prevent you from making it to class or exams, please obtain a doctor's statement certifying your circumstance.

Academic Integrity
As a student at Purdue you are subject to the Purdue University Student Code of Conduct, which enjoins you to respect the highest standards of honesty and integrity. All work that you submit in this course must be your own; unauthorized group efforts are considered academic dishonesty. See the online brochure Academic Integrity: A Guide for Students for definitions and sanctions. Academic dishonesty is a serious offense which may result in suspension or expulsion from the University. In addition to any other action taken, such as suspension or expulsion, a grade of F will normally be recorded on the transcripts of students found responsible for acts of academic dishonesty. Students are encouraged to report academic dishonesty to the instructor directly, or to the Office of the Dean of Students.

You may discuss assignments in a general way with other students, but you may not consult anyone else's written work. Among other ways to get an F, you are guilty of academic dishonesty if:

- You examine another student's solution to a written assignment
- You allow another student to examine your solution to a written assignment
- You fail to take reasonable care to prevent another student from examining your solution to a written assignment and that student does examine your solution. For example, if you allow another student to check his/her email from your terminal while you step out of the room, you have failed to take reasonable care to prevent him/her from accessing your files.

Do not con yourself into thinking that you can hide any collaboration. The risk of getting caught is too high, and the standard penalty is way too high.

If we find reason to believe that a student has cheated on any assignment, we may inform the student promptly, or we may decide to silently accumulate evidence against the student on later assignments.
Grading
Final grades will be assigned according to the following approximate weighting:
Final Project Score – 70%
Exam – 25%
Homework and Quizzes – 5%

The grading breakdown for the project is as follows:

- Project Charter – 5%
- Requirements Document – 15%
- Design Document – 15%
- Sprint 1 Planning Document – 7%
- Sprint 1 Review – 10%
- Sprint 1 Retrospective – 3%
- Sprint 2 Planning Document – 7%
- Sprint 2 Review – 10%
- Sprint 2 Retrospective – 3%
- Sprint 3 Planning Document – 7%
- Sprint 3 Review (Final Project Presentation and Demo) – 15%
- Sprint 3 Retrospective – 3%

These values summed represent the project score.

Peer Evaluations and Final Project Score
The project score will be subject to a multiplier based on peer evaluation by your team members. Each person will evaluate his or her team members by distributing $10 \times (n - 1)$ points, where $n$ is the number of team members, among them. For a given member, a multiplier will be determined by summing the points given to them and dividing by $10 \times (n - 1)$. This procedure will be done twice during the semester. The average of the resulting two multipliers, capped at 1.15, will then multiply the project score to generate the final project score. A deadline will be given for each aforementioned evaluation. Failure to meet the deadline will result in a decrease in your final multiplier by 0.05 each day past the deadline.

Questions and Answers
Questions of general interest should be posted on the course piazza site. Answers will be posted as soon as possible. Project questions should be directed to the appropriate project coordinator via email. Answers will be sent to you directly. If you need to contact a specific TA or instructor, send email to that individual or go see him/her during office hours.

Modifications
This syllabus may be modified at any time with notification.

*** As an interesting side note, a significant portion of this syllabus is copied from Dr. Dunsmore, Dr. Hosking's, Dr. Brylow's, and Dr. Hu's policy pages from previous semesters. One of the major differences between plagiarism and proper reuse is giving credit where credit is due. ***