CSE 360 Introduction to Software Engineering
Fall 2015

Dr. Debra Calliss
debra.calliss@asu.edu

BYENG 524
Office hours: Monday, Wednesday 10:00-12:00

Course Description: Software lifecycle models; software specifications and standards; team-based software development and project management; testing and quality assurance; legal and ethical issues. Prerequisites: CSE 240 or CSE 220

Text: Object-Oriented and Classical Software Engineering, Stephen Schach, 8th Edition, 2010, ISBN: 978-0073376189, McGraw-Hill Higher Education.

Course Goals: Students who complete this course will

1. Understand and apply basic software engineering techniques and approaches covering requirement, design, coding, testing, and maintenance within a process model
2. Develop an awareness of national and international standards
3. Be able to follow code of ethics
4. Be able to elicit and model requirements
5. Be able to use software architecture models
6. Be able to create analysis and design models
7. Be able to develop code
8. Be able to test code
9. Be able to plan and track software development activities
10. Be able to develop software artifacts using current software engineering tools
11. Be able to conduct software quality assurance
12. Be able to document and evaluate software product artifacts and team activities
13. Be able to present software engineering activities and products in oral and written forms

Topics:

- Software lifecycle models
  - Use of common software engineering process models
  - Strengths/weaknesses of software engineering process models
  - Awareness of software development standards and process improvement practices
  - Software engineering code of ethics
- Project management
  - Planning and tracking a small-scale software development project
  - Software risk management techniques
  - Awareness of software improvement processes
- Software development methods
  - Component-based software analysis and design models including structure and behavior

    - Component-based code implementation and documentation
    - Classical analysis and design principles
    - Awareness of trends in modeling methods
- Software tools for team-based software engineering
    - Analysis and design tools with support for standardized modeling techniques
    - Programming environments that automate parts of program construction process
    - Code management with version control
- Quality assurance
    - Unit, integration, validation, and system testing
    - White-box and black-box testing techniques
    - Software inspection including code reviews

Software Tools: You will be using several software development tools in this class including the versioning software GitHub (http://github.com), and a suitable IDE such as Eclipse (http://eclipse.org). The programming language required for the course is Java.

Work breakdown:

- 2 midterm exams 30% (Scheduled for Sept 21 and Oct 26) *
- 1 final exam 20% (Scheduled for Dec 7, 12:10PM) *
- In-class exercises 10%
- Homework 40%

* You must average at least 60% on the exams to receive a grade of C or higher in the class.

More information

- Exams are closed book/notes. Questions range from definition, multiple choice and short answer. Content is based on lectures, assigned reading and homework assignments.
- There is a seating chart for each exam. You must sit in your assigned seat and provide a picture id for exam submittal.
- Two in-class exercises are dropped in calculating your scores.
- Homework consists of both individual and group activities.
- Semester grades are based on 90+ = A, 80+ = B, etc. Plus or minus grades may be given.

<u>Academic Integrity</u> You are expected to do your own work and act in a professional and ethical manner. **<u>Plagiarism or misrepresentation is not acceptable in any form.</u>** Academic integrity violations are given the maximum penalty such as failure of the class and removal from the program.