**Department of Computer Science**

**Spring 2019**

**Course Number and Title**

        CS4800 Software Engineering

**Instructor**

       Mohamed Ibrahim

       Office:  Bldg. 8 Room 45

       Office Hours: TuTh 8:15 - 9:15 PM

       E-mail:  [moibrahim@cpp.edu](mailto:moibrahim@cpp.edu)

**Lecture Meeting**

       Time:  TuTh 7:00 - 8:15 PM

       Place:  Building 8, Room 348

**Textbooks**

1. Ian Sommerville, *Software Engineering, 7th Edition*, Addison-Wesley, 2006.
2. Roger Pressman, *Software Engineering: A Practitioner's Approach, 6th Edition*, McGraw-Hill, 2005.

**References**

1. Stephen Schach, Object-Oriented & Classical Software Engineering, 5th Edition, McGraw-Hill, 2005.
2. Eve Andersson, Software Engineering for Internet Applications, MIT Press, 2006
3. Alan Koch, Agile Software Development, Auerbach Publications, 2004
4. John W. Horch, Practical Guide to Software Quality Management, 2nd Edition, Artech House, 2003
5. Valentino Lee, Mobile applications : architecture, design, and development, Prentice Hall, 2004
6. Nicolai Josuttis , SOA in practice, O'Reilly, 2007
7. Mark Endrei, Patterns: service-oriented architecture and web services, IBM Corp., 2004

**Prerequisite**

CS 3310 with a grade of C or better.

**Course Objectives**

Upon the completion of this course the students shall acquire the following knowledge

and skills:

* Know the history and current status of software production and the scope of software engineering practice.
* Explain the software lifecycle and models of software process (such as Spiral, TSP, RUP, Scrum, or XP).
* Comprehend the models, techniques and software tools for project planning and management (such as IDEs, defect tracking systems, version control systems, or estimation tracking systems).
* Learn models for estimating software cost and deadlines (such as function points, story points, COCOMO, or wide-band delphi).
* Be aware of the differences among various methodologies for software development.
* Gain experience in the principles and techniques of effective user interface design
* Use the techniques for capturing user requirements and specifying software (such as contextual inquiry, user stories, scenarios, or formal specifications).
* Design and architect software systems using design principles (such as cohesion, coupling, abstraction, or information hiding) and design patterns.
* Read and create design diagrams (such as class diagrams, state charts, object diagrams, or dataflow diagrams) by hand and with tool support.
* Take a specification or a design diagram of a large system and implement a feature set.
* Appreciate software quality assurance techniques (such as white-box testing, black-box testing, regression, unit tests, program analysis, or inspection).
* Acquire experience in team-oriented software development

**Course Description**

Models of the software development process and metrics. Software requirements and specifications. Methodologies, tools and environments. Human-computer interaction. Software architecture, design and implementation techniques. Project management. Cost estimation. Testing and validation. Automated build, deployment and continuous integration. Maintenance and evolution.

**Exams**

There will be two exams. The exam dates will be announced in the class and posted on the course website.

**Homework Assignments**

There will be several assignments, each will be due at the beginning of class on the due date. There will be no substitution work for any assignments. The following rules are applied on any late homework assignments:

* Late submission will be penalized 5% per day late.
* Saturday and Sunday are counted as  “regular days” when it comes to calculating late penalties.
* Late submission will not be accepted after solutions are handed out or illustrated in class.

**Quizzes**

There will be several scheduled quizzes.

**Team projects**

Students work in small groups and participate in all the development phases (requirements analysis, design, construction, testing and documentation) of a software system. As well, each group has to address the control of the development process by constructing and following a detailed software development management plan.

**Grading**

      Test#1                                                                                          20%

      Test#2                                                                                          20%

   Quizzes/ In-class exercise/Class participation 20%

      Homework Assignments                         20%

Team projects 20%

**Letter Grades Scale**

**A      >=90**

**86  <=   B+   < 90**

**80  <=   B     <=85**

**76  <=   C+   <80**

**70  <=   C     <=75**

**66  <=   D+   < 70**

**60  <=   D     <= 65**

**F      <60**

**Class Policies:**

1. Participation is an integral component to complete this course satisfactorily.

2. Late submission of assignments or make up for the exams is **only** accepted in case of **documented** emergency situation.

3. Please check your **Cal Poly email account** and **Blackboard** every day for course announcements.

4. All submitted works in the course must be your own. Academic dishonesty will be handled according to university policy. The consequence of academic dishonesty is an immediate F grade in the course at the very least.

5. Attendance is expected at all class sessions but not graded. Students are responsible for all material presented in the course whether or not they attend the class, including announcements about course procedures. Office hours cannot be devoted to cover missed class lectures.