Software Engineering

Fall 2015, Sul Ross State University

Instructor: Dr. Kennard Laviers

Office Location: ACR 107 **Office Phone**: 432-837- 8500

Email: kennard.laviers@sulross.edu

Office Hours:

M 8:30-9:00 & 1:30pm - 2:30pm
T 8:30-9:30 & 2:00pm - 2:30pm
W 8:30-9:00 & 1:30pm - 2:30pm
R 8:30-9:30 & 1:30pm - 2:30pm
F 8:30-9:00 & 1:30pm - 2:30pm

Class: *MWF* 10:00 pm - 10:50 am MAB 302; **Lab:** *M* 2:00 pm - 2:15 pm. BAB 302

Program Learning Objective

1. Understand the fundamental concepts of computer science including algorithms and data structures.
2. Understand modern computer systems, databases and networking.
3. Display an understanding and ability to implement current programming methodologies.
4. Become proficient with systems design based on object-oriented programming.
5. Work as a team in workgroup environments.

Textbook: Software Engineering (9th Edition) Hardcover **use pre formatted date that complies with legal requirement from media matrix** – March 13, 2010
by Ian Sommerville (Author)

Publisher: Pearson; 9 edition (March 13, 2010)
Language: English
ISBN-10: 0137035152
ISBN-13: 978-0137035151

http://www.sulrossbookstore.com/CourseMaterials.aspx
Links: http://www.compileonline.com/
Online Tools: Google Doc's Draw is a free app offered by Google.

Course Objectives

This course will introduce students to fundamental software engineering techniques. Students will learn best practices and design using the Unified Modeling Language (UML) and will participate in developing a large application throughout the semester. Students will learn how to define Software Requirements, perform a feasibility analysis, create Class Diagrams, State Charts, and Communications Diagrams as part of the Software Design process.

Specific topic coverage and schedule:

Week 1

Subject: Course Overview

Goal: Allow students to understate the full scope of this course and what the expectations are for successful completion of the class.

Learning Objective:

1. Students will know what the difference is between software engineering and computer science.
2. Students will know the attributes of good software.
3. Students will understand when it is important to use the engineering process.
4. Students will have a basic understanding of how ethics plays a role in any engineering process.

Notes:

Reading: Chapter 1

Week 2

Subject: Programming Overview Part I

Goal: To ensure all students are able to program at a basic level needed to successfully complete project assignments in this course.

Learning Objective:

1. Students will know how to implement identifiers, work with basic input/output functions, for and while loops, and know how to construct the basic structure of a typical object oriented programming language.
2. Students will be able to perform basic Boolean math.

Notes:

Week 3

Subject: Programming Overview Part II

Goal: To ensure all students are able to program a basic user interface.

Learning Objective:

1. Students will know how to implement a graphic interface using Unity.
2. Student will be able to integrate buttons in an application
3. Students will be able to implement dynamic labels and input fields in their applications
4. Students will be able to implement Toggles
5. Students will be able to implement Scroll lists
6. Students will be able to import and use sprites in the interface
7. Students will understand how to use anchors in their interface

Notes: This module will require a significant amount of self study if the student has not done any graphic programming before. Student will be provided links to additional tutorials to aid them.

Reading:

Week 4

Subject: Object Oriented Programming

Goal: This module is designed to ensure students understand object oriented programming will enough to be able to effectively design the systems using UML which is based on object oriented development.

Learning Objective: Students will understand and be able to effectively implement:

1. Classes
2. Member Variables
3. Methods
4. Constructers
5. Protection levels
6. Static vs. Dynamic Classes and Types
7. Abstraction
8. Overriding methods
9. Interfaces

Notes: The goal for this module should be met by taking the object oriented programming class but this should serve as a nice refresher if the student is rusty in this area.

Week 5

Subject: Software Engineering Overview

Goal: This module will introduce Software Engineering and two popular approaches to engineering a large scare project. The first is the first and oldest approach known as the waterfall method and the next is the most common approach used today know as the evolutionary or iterative approach.

Learning Objective:

1. Understand key differences between a waterfall approach and evolutionary approach.
2. Understand why the waterfall method is hardly ever used anymore
3. Understand the parts of waterfall method
4. Understand and memorize the parts of the iterative method
5. Understand the advantages of the iterative approach

Notes:

Reading: Chapter 2

Week 6

Subject: Making the Software Requirements Document

Goal: This section teaches students how to effectively perform a feasibility study for an application and generate a feasibility report. Additionally the students are taught effective methods to illicit detailed requirements from a user and generate a requirements document.

Learning Objective:

1. Be able to generate a feasibility study
2. Effectively acquire detailed requirements from the user
3. Successfully generate a requirements document

Notes:

Reading: Chapter 4

Week 7

Subject: Introduction to Software Design and UML

Goal: This module will introduce the tools necessary to build a comprehensive software design using the Unified Modeling Language (UML)

Learning Objective:

1. Gain a Basic understanding of Use Case Diagrams
2. Gain a Basic understanding of Class Diagrams
3. Gain a Basic understanding of State Diagrams
4. Gain a Basic understanding of Communications Diagrams
5. Gain a basic understanding of design patterns

Notes: This model introduces the basic concepts of UML that will be explained in greater detail later weeks.

Reading: Chapter 5

Week 8

Subject: Use Case Diagrams

Goal: Students will learn how to identify system use cases and property document each case as a written scenario and will also diagram the use cases with UML.

Learning Objective:

1. Students will be able to properly identify use cases from a requirements document.
2. Students will be able to document a use case
3. Students will be able to diagram a use case using UML

Notes: Often use cases are included in the requirements document so this module revisit the requirements document to add the use cases.

Week 9

Subject: Class Diagrams

Goal: Students will be shown how to identify nouns in a requirements document that make good candidates for Classes and Objects in the system. Students will be introduced to techniques to also extract actions and how to determine where the functionality should be placed. Finally, the students will be taught how to diagram all the classes and interactions between classes using UML.

Learning Objective:

1. Ability to identify candidate classes
2. Identify functions and what classes they should go to
3. Be able to diagram abstraction, aggregation and class associations

Notes: This is the most important module on UML and may extend to 2 weeks

Week 10

Subject: State Diagrams

Goal: State Machines are a critical part of computer science. This module will provide a detailed understanding of finite state machine (FSM) theory and how to identify software system states and how to diagram system state-machines.

Learning Objective:

1. Ability to identify states in a proposed software system
2. Ability to successfully design a state chart using conditions, actions and attributes
3. Have a basic understanding of state-based controllers and how they would work in a very complex software system

Notes:

Week 11

Subject: Communication and Sequence Diagrams

Goal: This module will introduce students to properly identify requirements for message passing in a software system and how to diagram those communications.

Learning Objective:

1. Understand communications between systems or modules in a software system
2. Know the difference between a communication's diagram and a sequence diagram
3. Be able to design a sequence diagram from a requirement

Notes:

Week 12

Subject: Detailed Design Document and Implementation

Goal: In this section the Class diagram (Architectural Diagram) designed earlier will be extended out to a full detailed design document that is ready for implementation.

Learning Objective:

1. Student should be able to properly identify all functions and member variables of each Class in a software system
2. Student should be able to properly identify input and output requirements of all functions in each class
3. Student should be able to write-up a detailed design for each class of a system

Notes:

Reading: Chapter 7

Week 13

Subject: Testing

Goal: Students will be provided an overview of typical system testing techniques

Learning Objective:

1. Student will be able to list the various types of testing and have a basic understanding of each.
2. Students will be able to draft a basic testing document

Notes:

Week 14

Subject: Unity Testing

Goal: Student is introduced to basic techniques to test system unites including edge-case testing, golden path testing and exhaustive testing.

Learning Objective:

1. Students will be able to identify what type of testing is being used when giving a description of testing that is taking place.
2. Students will be able to describe how unit testing fits in with the over-all system testing and why it is important.

Notes:

Reading: Chapter 8

Week 15

Subject: System Integration and Testing

Goal: Students are shown how everything fits together and the final application is completed and tested.

Learning Objective:

1. Student will be able to implement the detailed design document
2. Student will be able to list the basic types of system testing
3. Student will be able to follow a test document and understand its importance

Notes:

Reading: Chapter 11

Attendance
Any student who accumulates 10 **unexcused** absences (MWF Classes) or 7 **unexcused** absences (MW classes) will be automatically dropped from this course.

Need for Assistance
Qualified students with disabilities needing academic or other accommodations to ensure full participation in the programs, services and activities at Sul Ross State University should contact the Disabilities Services Coordinator, in Counseling and Prevention Services, Ferguson Hall 112, Box C-117, Alpine, Texas 79832. Please notify me before the third day of classes.

Course Policies
Quizzes and assignments must be submitted on time. I have set up rules in Blackboard so that assignments cannot be submitted after the due date.

Academic Dishonesty: Honesty in completing assignments is essential to the mission of the university and to the development of the personal integrity of the student. Cheating, plagiarism, or other kinds of academic dishonesty will not be tolerated and will result in appropriate sanctions that may include failing an assignment, failing the class, or being suspended or expelled. Suspected cases in this course may be reported to Student Life.

Posting of Grades
As soon as assignments, exams, and quizzes are graded, the grades will be posted in Blackboard.

Grading
Letter grades will be determined using a standard percentage point evaluation as outlined below. Please note that this is a tentative schedule and can change. Any changes that happen will be updated in Blackboard. Due Dates for assignments will also be posted in Blackboard.

Your final grade will be determined by calculating points based on the following weights:

- A 90 - 100 points
- B 80 - 89 points
- C 70 – 79 points
- D 60 – 69 points
- F below 60 points

More resources

Book Web Site:

http://ifs.host.cs.st-andrews.ac.uk/Books/SE9/

Videos <SE10>

http://iansommerville.com/software-engineering-book/videos/

UML 2.0 Tutorial

http://www.sparxsystems.com/resources/uml2_tutorial/index.html

http://www.agilemodeling.com/essays/umlDiagrams.htm

YouTube

ALL UML Diagrams

https://www.youtube.com/watch?v=OkC7HKtiZC0&list=PLGLfVvz_LVvQ5G-LdJ8RLqe-ndo7QITYc

1 (Intro and Use Case)

https://www.youtube.com/watch?v=OkC7HKtiZC0

2 (Activity Diagrams)

https://www.youtube.com/watch?v=XFTAIj2N2Lc

3 (Class Diagrams)

https://www.youtube.com/watch?v=3cmzqZzwNDM

4 (Sequence Diagrams)

https://www.youtube.com/watch?v=cxG-qWthxt4

5 (Communications Diagrams)

https://www.youtube.com/watch?v=TL4ABTx_RtE&list=PLGLfVvz_LVvQ5G-LdJ8RLqe-ndo7QITYc&index=5

6 (Timing Diagram)

https://www.youtube.com/watch?v=F3yROOhK8qk&list=PLGLfVvz_LVvQ5G-LdJ8RLqe-ndo7QITYc&index=6

7 (Component Diagrams)

https://www.youtube.com/watch?v=KQUGFFN4M90&index=7&list=PLGLfVvz_LVvQ5G-LdJ8RLqe-ndo7QITYc

8 (State Machine)

https://www.youtube.com/watch?v=_6TFVzBW7oo

9 (Deployment Diagrams)

https://www.youtube.com/watch?v=nTtQwGoUUNc&index=9&list=PLGLfVvz_LVvQ5G-LdJ8RLqe-ndo7QITYc