|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Spring 2020 **CSCI 463**: Introduction to Software Engineering   |  |  | | --- | --- | |  | | | **Instructor** : | **Dr. Hassan Reza** | | **Email** : | **reza at aero.und.edu** | | **Phone:** | **701/777-4135** | | **Office hours :** | **T,TH: 1:00 to 1:50 or by Appointment** | | **Class meets :** | **T,TH: 3:30 to 4:45;** | | **Room** : | **Lenard 110** | | **Credit hours :** | **3** | |
|  |  | **Source:**   * **Required: Text:** Software Engineering, Theory and Practice, 4th edition, Shari Lawrence Pfleeger and Joanne M. Atlee, 2010 Prentice Hall. * **Reference Text**:  Engineering Software Products: An Introduction to Modern Software Engineering, by Ian Sommerville, Pearson, 2020; * **Possible References**: Journals and/or Conference Proceeding Articles   **Introduction:**  Successful development of software products is no accident. It depends on an in-depth understanding of the process (i.e., set of activities and practical methods) by which the software is constructed. Software engineering is a fast moving scientific area that was originally coined by the computer science pioneers such as Bauer and Dijkstra in the late 1960s as the establishment and use of sound engineering principles in order to construct software products that are reliable and efficient.  Software engineering has come a long way since 1960's. Currently, it is considered as an engineering discipline, which combines the ideas from engineering, management, and math disciplines in order to improve our ability to build complex software systems on time and within the budget.   **Course Description:**  CSci 463 is an introduction to the fundamental aspects of software engineering. Emphasis in this course, for the most part, is placed on the initial phases of the software development life cycle, namely, design and analysis of software intensive systems.  **Course Prerequisites and Prerequisite Fulfillment:**  Prerequisite for this course is successful completion (grade C or better) of CSci 242 (Data Structure) and CSci 363. This course demands both independent and cooperative work. Working independently requires that you have good time management skills and self-directed learning skills. There are no prerequisites for team work other than a willingness and flexibility to participate and learn as part of a team. This course is an element of software engineering cluster.  **Course Objectives:**  This course is important for anyone involved in the construction of a software product. This course, among other things, offers   * [Good employment opportunities;](http://money.cnn.com/video/news/2012/01/05/n_computer_science_jobs.cnnmoney/) * Development of interpersonal and group working skills; * Good starting point for a professional career in software engineering.   Students will learn how to deliver satisfactory software products systematically. More specifically, this course will explore key elements and principles that are used to define software engineering: Abstraction and Refinement, Analysis and Design methods and Notations, Modularity and Architecture, Software life cycle and Modeling process, Software reuse and component technology, and Design Patterns. We also discuss the difference between computer science and software engineering and what we mean by "high quality" software product. Upon successful completion of the course, the student should be able to understand the role and components of software engineering methodology. Possible topics included (but not limited):   * Software Engineering Principles * Software Process Models * Requirements Engineering * Design Specification * Software Architecture (high level design) * Software Design (low level design) * Software Implementation (code writing) * Software Verification and Validation (Testing) * Management of Software Engineering (planning and managing the project).   **Course Procedures:**  The course will utilize lecture and discussion, as the main teaching tools for presenting the course material. Students will be expected to read the material and be prepared to discuss the readings and the assignments in class.  **Possible Advanced Courses in Software Engineering:**   * **Formal Methods in Software Engineering** (CSci 562):   + A study of formal specification in the software development process including operational, denotational, and object oriented techniques; * **Software Architecture** (CSci 582):   + A study of formal and semi-formal notations used to specify communicational and computational elements of a complex system; * **Advanced Software Engineering** (CSci 565):   + This course covers the entire software development life cycle. Emphasis is placed on advanced topics including prototyping, verification and validation, formal methods, software configuration management, etc.   **Course Grading Policy:**   * Two Exams 50% (Exam 1: 25% and Exam 2: 25%); * Assignments and Pop Quizzes (10 %); * Team Project   + Under graduate students (40%)   + Graduate students (30%) * **Graduate Students Extra Work (10%)**   **Exams:**Exams are one hour-long and none comprehensive. The exams will contain questions covering material in the text (required and references), homework, and the lecture notes. There are two exams as follows:   * **Exam 1: March 3th, 2020** * **Exam 2: April 28th, 2020**   **Make-up Exam and Quiz Policy:**No make-up exams or quizzes will be conducted except for documented illness (i.e., a letter from your physician) or personal emergency.  A student not taking an exam (or quiz) will receive an F for that exam (or quiz).  **Assignments and Pop Quizzes:** There will be several assignments and quizzes throughout the course. The topics of these assignments will be taken either from the textbook or from other resources relevant to the course and will be announced in advance. Assignments should be turned in the class. For every day of late submission 20% of the points will be deducted. You are advised to hand in assignments in class. Any homework placed without my knowledge (e.g., in my office mail box) will be accepted without any warranty. In case an assignment is lost, it is your responsibility.  Recommendations are: to save everything, retain important papers and computer files until the course is finished.  There will also be pop quizzes on material we have recently read/discussed in class. The quiz will normally be one or two basic questions. Your answer will be a given S (Satisfactory) or U (Unsatisfactory). There will be 8-10 quizzes for the course. To accommodate those who miss class, two low quizzes for each student will be dropped.  **Submitting Assignments:** All programming assignments (if any) must be submitted either by email, on the disks/CDs, or hardcopy. The following guidelines are used as the grading criteria:   1. If the programming assignment cannot be compiled due to compilation errors, the assignment will receive **a zero point**; 2. If the programming assignment compiles with warnings, the assignment may lose up to 100% of your grade depending on the nature and the severity of the warnings; 3. You program must adhere to proper documentation and programming styles (e.g., comments); 4. Your submission must include both the source code and executables file(s); 5. All the work is submitted to the **UND Blackboard systems**. The submission must include: your name, assignment number, and class in the subject heading (e.g., **John-Smith.HW1.Csci463**).   ***Note 1: All non-Programming assignments must be typed and spell checked otherwise they won't be graded.***  I normally return assignments, exams, and quizzes to the class. If you are absent, it is your responsibility to pick up your work at my office or inform by emails.  **The Team Project:**  There is one **Modest Software Engineering Project (MSEP)** for the course, which counts 30% - 40 % of course grade. The project is done in groups (2-3 people).   For the project, you and your team members need to analyze, specify, design, implement, test, and demonstrate a typical information system.  Due to time limitation, you are not expected to deliver a full fledge marketable software product, but the end result must be a reasonable prototype. This prototype should be used as the basis or proof of concept for defining or developing a real world product that can be funded by potential investors.  Your grade on the project will be determined by the quality of your work on each milestone. Should the grade on a milestone be below a B or C, you can resubmit that milestone one week later for a re-grade (for a maximum grade of B or C). The final delivery is **NOT re-graded** because of time constraints. The project consists of four deliveries and the final presentation as follows:   1. Cost Estimation, and Project Plan Documentation: **~~February 17~~; February 21** 2. Software Project Definition and Specification, Cost Estimation, and Project Plan Documentation: **March 24** 3. Software Design Documentation (SDD): **April 23**; 4. Software Test Documentation (STD), and Source Code (hard copy and CD):  **May 14**; 5. Project Demonstration in Class by Project leaders/members: **May 5th and 7th**.   **Team and Project Mgt:** It is highly recommended that each team develops some type of organizational structure by assigning tasks and responsibilities to individuals, identifying a reasonable mechanism to measure progress, proper means for making decisions and communicating the issues.  According to **Martin Reddy**, Software Engineering Manager at Apple, “one of the biggest challenges of many careers is making the transition to management (leadership) and having to successfully lead teams of people. Often team problems are more difficult to solve than technical ones, but they can also be immensely more rewarding to solve because you have the power to improve people’s lives”. Therefore, it is highly recommended that you select a member as a team leader to lead the team as soon as possible. A team leader must be fair and trustworthy. S/he must possess, at least, the following qualification and must willing to perform the following tasks:   * **Qualifications:**   + Good communication skills   + Good management skills   + Good problem solving skills   + Availability and flexibility * **Responsibilities:**   + Lead the team for project development   + Schedule the meetings   + Conduct project reviews   *This is a team project, and you will be graded (or judged) as a team. All team members are responsible for the proper working of the team. Team leaders/members are expected to be in class for the project demonstrations on May 2-7, 2016. Failure to attend any presentation may affect the final grade for the project.*  ***Graduates students extra work:*** Graduate students taking 400-level courses are required to do extra work by the School of Graduate Studies. Extra work constitutes 10% of the final grade. An example of Extra Work includes a publishable paper/report addressing a research problem in any area of the Software Engineering. The report should adhere to ACM/IEEE template.  **Grading Scale for ALL students:**   |  |  | | --- | --- | | **90 - 100%** | **A** | | **80 - 89%** | **B** | | **70 - 79%** | **C** | | **60 - 69%** | **D** | | **below  60** | **F** |   **Participation & Attendance Policy:** Regular classroom attendance is required (this includes any team meeting held class time and outside of class time), but does not count as participation. Active participation in class involves preparing ahead of time, asking intelligent questions that the book does not answer, answering questions in class, and participating in classroom discussions. If you know you will miss a class ahead of time, email the instructor. Make friends with your classmates so they will be willing to share notes, which problems were assigned, etc., for when emergencies arise and you do miss class.  **Civility in the Classroom:** Students are expected to assist in maintaining a classroom environment, which is conducive to learning.  Please do not bring food to the classroom and always turn your cell phone (ringer) off if you have to have it with you.  **Disability Conditions:** If you have a disability condition that may affect your participation in this course and wish to discuss academic accommodations, please inform me as soon as possible.  **Drops & Withdrawals:** Students should not assume that the instructor will administratively drop the student for non-attendance. Withdrawal from the course is the responsibility of the student. Should a student decide to withdraw from the course, simply not attending class sessions, rather than formally withdrawing from the course, may result in an F on the student's transcript.  **PLAGIARISM, PROFESSIONAL ETHICS & PERSONAL INTEGRITY:** *The UND academic policy clearly states an expectation that each student MUST submit his/her own work do otherwise, that is, to submit anther's efforts as one's own, is at its least plagiarism, and at its most unethical and without integrity. Indeed, it is a matter of professional ethics and personal integrity to graciously accept accolades for a job well done, as well as to publicly acknowledge those to whom partial credit is owed.* *Those found to be cheating on exams, assignments, Project, etc. will receive an* ***F*** *for the* *COURSE, and the matter will be discussed with the appropriate Dean and/or Chair. I do not* *anticipate any problem here.*  ***Disclaimer: This syllabus is intended to suggest the outline of the course; it is not absolute. Changes to the syllabus, should they occur, will be announced in class.***  ***Last Update Jan.9.2020   by H.R.*** |
|  |  |  |