COURSE: SE 501 - Introduction to Software Development  
DATE: Fall 2020 (August 17, 2020 - November 25, 2020)  
INSTRUCTOR: P. M. Jackowitz  
OFFICE: LSC 192  
OFFICE HOURS: As posted (office door and online), and by appointment.  
TELEPHONE: (570) 941-6107  
EMAIL: paul.jackowitz@scranton.edu  
WWW: http://www.cs.scranton.edu/~jackowitz

CATALOG COURSE DESCRIPTION: (Prerequisite, admission to the program) This course serves as an introduction to the discipline of Software Engineering, involving both a study of theory and practice. Significant ideas and developments are emphasized along with an examination of terminologies, classifications, paradigms, and methodologies. This course also provides an opportunity to review essential computer science material (data structures, programming languages and environments, systems and architectures) as appropriate within this context.

Student Learning Outcomes: Upon completion of this course, a successful student will have the ability to do each of the following:

- Provide an overview of the history of Software Engineering.
- Identify some of the prominent individuals who have made significant contributions to the development of Software Engineering and explain their contributions.
- Provide an insightful definition for Software Engineering and be able to substantiate an opinion on whether or not Software Engineering should be considered an "engineering discipline" on par with the traditional areas of engineering.
- Define and discuss at length the Software Life Cycle.
- Identify and explain numerous Software Process Models, discuss their respective advantages and disadvantages and identify situations that merit their application
- Define and discuss at length Requirements Analysis, Design (at both the Architectural level and the Modular level), Programming, Testing and System Maintenance
- Explain the role and importance of Project Planning, Project Management and metrics in Software Engineering
- Be familiar with the sources of research in Software Engineering and be able to read, understand and evaluate the significance of current developments as they pertain both to the theory and practice of Software Engineering
- Write clearly and in an informed way on topics of interest in Software Engineering
- Function as a contributing member of a Software Development team in a variety of capacities, including but not limited to, requirements gathering, architecture design, modular design, programming, testing and software maintenance.


REFERENCES: As presented and as required.

(This site serves as our primary electronic communication tool for this course. You will use it to access required and optional course material and to submit and review assignments. Initially, you must register to obtain full access to the functionality of this site.)
GRADING:  Tests: (approximate date)          Worth
                      September 28th                  25%
Final Exam: (Comprehensive, as scheduled)  25%
Assignments/Homework                        30%
Reports/Presentations/Participation         20%

(Your attendance at all classes is required, and your participation is essential. Thus, attendance and participation will receive strong overriding consideration when determining grades.)

PROCEDURES:
Note that in light of the ongoing COVID-19 pandemic, the University has established specific behaviors for all persons on campus. We are all expected to be informed of these and to adhere to them.

Lectures:
- please sit in the same seat for every class
- feel free to ask and answer questions, and to contribute to discussions
- classroom use of electronic devices/gadgets (including computers) is at the full discretion of the instructor. (Distracting yourself or others will not be tolerated.)

Tests and Quizzes:
- always announced in advance, and no make-ups will be given
- notice must be given if a test must be missed due to serious illness or emergency

Assignments:
- each student is required to do his/her own work on each assignment
- discussions and mutually beneficial collaboration among students is encouraged, but must not be to the point of representing someone else’s effort and understanding as your own as this would be considered to be academic dishonesty (See Academic Code of Honesty in the Student Handbook at http://catalog.scranton.edu/mime/media/view/42/5075/2018-2019-Student-HandbookFINAL.pdf)
- academic dishonesty will be dealt with severely
- each assignment will have a specified due date, and a specified deadline
- normally the deadline is later (typically, but not always two days) than the due date
- work submitted after the due date is considered to be “late”, will be accepted for grading but may be assessed a penalty (depending upon how late it is, and whether or not worthwhile preliminary work had been submitted by the due date).
- work may not be submitted after the deadline; it is considered to be too late, may not be accepted for grading, and may receive a grade of zero (depending upon whether or not worthwhile preliminary work had been submitted prior to the due date or prior to the deadline).
- incomplete work generally will receive a grade much higher than zero
- work not submitted will receive a grade of zero

Other:
- Obligations: https://www.scranton.edu/equity-diversity/faculty-resources.shtml
List of Topics:
1. Definitions of Software Engineering
2. History of Software Engineering
3. Writing Programs vs Building Systems
4. Evaluating Success and Measuring Quality
5. Traditional Software Process Models
6. Agile Software Process Models
7. Requirements Analysis
8. Architecture
9. Design
10. Testing and Quality Assurance
11. Miscellaneous Topics