COURSE TITLE: Software Engineering II
COURSE CODE: COMP 3613
TYPE: Core
LEVEL: 3
SEMESTER: 1
START DATE: SEPT-01-2016
DEPARTMENT and FACULTY: DCIT/FST
CREDITS: 3
PRE-REQUISITE(S): COMP 2606

ESTIMATED STUDY HOURS:
2 1-hour lectures, 1 1-hour tutorial, 5 hours per week independent study

COURSE DESCRIPTION

This course is a continuation of developing skills surrounding software engineering, its principles and practical applications within the computer science curriculum. This course will expose students to the required engineering rigors of specifying, designing, developing and maintaining product-quality code. It will prepare them for the challenge of developing software systems as part of a team through a better understanding of development process methodologies, and an appreciation of the different challenges software engineers face in domains as varied as web-based systems, mission-critical systems and safety-critical systems.

COURSE RATIONALE

The ability to develop software utilizing agile methodologies is an integral part of the field of computer science or software engineering. Thus, this course explores advanced topics in software engineering, while ensuring that the students participate as developers and project managers during the software development cycle.

COURSE CONTENT

1. Software Engineering tools (version control, testing tools, continuous integration)
2. Project Management (Approaches, Planning and Resource scheduling, Effort Estimation, tools)
3. Coding practices (Defensive and secure practices, coding standards, integration strategies, software reliability/dependability)
4. Software Evolution (Software engineering in pre-existing code bases, software uses)

COURSE LEARNING OUTCOMES

Upon the successful completion of this course, the student will be able to:

1. Utilize an agile based methodology to create a sufficiently complex software product based on developed requirements.
2. Explain the impact of identified software evolution issues associated on the software lifecycle.
3. Explain the problems that exist in achieving very high levels of reliability

4. Apply techniques, coding idioms and mechanism when implementing software designs so that reliability, efficiency and robustness is achieved.
5. Build robust code using exception handling mechanisms
6. Utilize a selected coding standard in a small software project.
7. Demonstrate how version controls can be used to manage software releases through the application of a source code control tool in a small team-based project.
8. Develop communication skills through written material and oral presentations

COURSE ASSESSMENT

Assessment	**Learning Outcomes**								**Weighting %**	**Assessment Description**	**Duration**
	1	2	3	4	5	6	7	8			
Assignment 1	X	X	X						10	Research	
Assignment 2				X	X	X	X		10	Code	
Group Project	X	X	X	X	X	X	X	X	30	Documentation/Code	
Project Presentation							X	X	25	Rubric Based	
Individual Project Report								X	25	Rubric Based	
TOTAL %									**100%**		

TEACHING STRATEGIES

A project-intensive methodology for teaching software engineering is applied. The class work involved in this course directly complements the project work. Theoretical concepts are introduced through the case-studies. Students will work in groups on a software development project.

TEXTBOOK

Sommerville Ian A., Software Engineering 10th edition (2015), Pearson,

COURSE CALENDAR

Week	Topic
1	Lecture: Review
	Tutorial: Introductions
2	Lecture: Introduction to Agile Methodology
	Tutorial: Fundamentals of Scrum
3	Lecture: Project Given
	Tutorial: Fundamentals of Scrum
4	Lecture: Introduction to Agile Methodology Assignment 1 (Week 4)
	Tutorial: Code tools - git
5	Lecture: Agile Design
	Tutorial: Code tools - git
6	Lecture: Testing and verification
	Tutorial: Code practices – test cases (python)
7	Lecture: Testing and verification
	Tutorial: Code practices – test cases (python)
8	Lecture: Software Maintenance Assignment 2
	Tutorial: Continuous integration
9	Lecture: Software Maintenance
	Tutorial: Continuous integration
10 - 12	Software Evolution Project Report [Individual + Group] (Week 12) Tutorial: project support
13	Project Presentations