Course Specifications

General Information

Course name	Introduction to Software Engineering
Course number	ITSE2301
Faculty	
Department	
Course type	Major Needs
Course level	2
Credit hours (theoretical)	3
Credit hours (practical)	0
Course Prerequisites	

Course Objectives

1 - Student will be able to appreciate the importance of software engineering
2 - Student will be able to acquire knowledge of modeling software systems
3 - Student will be able to develop skills to model small software applications
4 - Student will be able to apply software engineering principles when developing software
5 - Student will be able to acquire knowledge of software architectural patterns
6 - Student will be able to give details of various software development processes, methods, and tools

Intended Learning Outcomes

Knowledge and Understanding	* Student will be able to explain the importance of software engineering * Student will be able to explain the challenges facing the development of software * Student will be able to describe software development processes, and their activities, and articulate the applicability and diversity * Student will be to explain causes of software change * Student will be able to explain the motivations behind the development of rapid software development processes * Student will be able to describe the principles of agile methods and practices of XP * Student will be able to discuss features and drawbacks of agile-based project management * Student will be able to describe requirements engineering process, requirements fundamental activities, and development methods * Student will be able to explain and apply requirements validation method * Student will be able to describe requirements change and management processes and related activities * Student will be able to describe architectural patterns, and their pros and cons
Intellectual Skills	* Student will be able to discuss the attributes of good software * Student will be able to differentiate between software engineering and other related disciplines * Student will be able to discuss professional responsibilities and ethical principles and issues related to the behavior of software engineer * Student will be able to describe the software development model, and their activities, and articulate the applicability and diversity * Student will be able to articulate reasons of diversity of software processes and software process models * Student will be able to discuss strategies to cope with software change * Student will be able to differentiate and relate principles of agile methods and the practices of XP * Student will be able to determine the similarities and differences between plan-driven and agile approaches to software development * Student will be able to distinguish between various types of requirements: user and system requirements, and between functional, non-functional, and domain requirements * Student will be able to explain and apply various system modeling approaches * Student will be able to describe class, object, sequence, component, use case, and package diagrams using UML
Professional Skills	* Student will be able to explain and apply test-first development * Student will be able to identify user, system, functional, non-functional, and domain requirements * Student will be able to drive user, system, functional, non-functional, and domain requirements * Student will be able to develop software requirements document * will be able to create system design of software

Course Contents

1 - Overview and Principles of software engineering. Process activities. Software requirements: Functional, non-functional, user and system requirements. Software design concepts. Software testing. Software evolution. Software stakeholders. Software process models: generic models and evolutionary. agile development methods such as XP and Scrum. Project management. System models: structural and behavioral. Software architectural: views, patterns and application architectures. Open source development. Test-driven development. User testing. Object oriented analysis using UML.

Teaching and Learning Methods

1 - Lectures

Students Assessment

Assessment Method	TIME	MARKS
Midterm I	6th Week	20
Midterm II	9th Week	20
Attendance & Assignments		10
Final Exam	16th Week	50

Books and References

Essential books	Software Engineering, by Ian Sommerville, 10th Edition, 2015