COMSATS University Islamabad
Department of Computer Science
Course Syllabus

Course Information:

Course Code: **CSE291**	Course Title: **Introduction to Software Engineering**
Credit Hours: **4(3,1)**	Lecture Hours/Week: **3**
Lab Hours/Week: **1**	Pre-Requisites: **None**

Catalogue Description:

This course introduces the different software process models by illustrating its phases and principles of software engineering. Topics include: Overview of Software Engineering; Software Process Models; Requirement Engineering Concepts; Software Design; Design Modeling; Software Quality Engineering;
Software Project Management; Software Maintenance and Software Evolution.

The main objective of this course is to construct a solid foundation for understanding and application of principles, techniques and technologies utilized in the development of good software systems by individual or teams. The objective of this course is to learn contemporary techniques to develop as well as manage the development of industrial strength software.

Text and Reference Books:

Textbooks:

1. Software Engineering: A Practitioner's Approach, Roger S. Pressman &Bruce R. Maxim, McGraw-Hill, 2020.
2. Engineering Software Products: An Introduction to Modern Software Engineering, Ian Sommerville, Global Edition, Pearson Education Limited, 2021.

Reference Books:

1. Software Engineering, Ian Sommerville, Pearson Education Limited, 2016.

Software Engineering with UML, Bhuyan Unhelkar, CRC Press, 2018.

Week wise Plan:

Lecture #	CDF Unit #	Topics Covered	Reading Material
1.	**1**	Orientation and general discussion of course structure. Grading policy and Assignments Introduction. Overview of Software Engineering Professional Software Development,and Software Engineering Practice. Software Engineering Ethics	Pressman: Ch1 Sommerville: Ch1
2.	**1**	Introduction to Software Development Life Cycle (SDLC) Adhoc Development Plan -driven and Agile Approach	Sommerville: Ch2
3.	**1**	Process Models: Waterfall Model and V Model	Sommerville: Ch2
4.	**1**	Evolutionary Development: Exploratory, Prototyping & Component-based Software Engineering.	Sommerville: Ch2

5.	**1**	Incremental and Spiral Process Model	Sommerville: Ch2
6.	**1**	Rapid Application Development ▪ Why we need to develop application earlier ▪ Structure of RAD ▪ Essential Aspects of RAD ▪ Application	Ref. Material
7.	**1**	Agile Development (Extreme Programming) ▪ Agile Software Development: ▪ What is agility? Agility & the Cost of Change, ▪ Agile Process, and Agility Principles. Extreme programming (XP)	Sommerville: Ch3
8.	**2**	Introduction to Requirements ▪ Definition ▪ Types of requirements o Functional & Non-Functional Requirements.	Sommerville: Ch4
9.	**2**	o Guidelines to specify requirements o Stakeholders Requirement Engineering Process	Sommerville: Ch4
10.	**2**	Requirement Elicitation Techniques/ Approaches	Sommerville: Ch4
11.	**2**	Case studies (to differentiate among requirements and how to write Functional and non-functional requirements of different system)	Ref. Material
12.	**3**	System Modelling: ▪ Context Models	Sommerville: Ch5
13.	**3**	Data Driven Modelling	Sommerville: Ch5
14.	**3**	Event Driven Modelling	Sommerville: Ch5
15.	**3**	Interaction Models: ▪ Usecase modelling	Sommerville: Ch5
16.	**3**	Practice Case studies	Ref. Material
17.		**Mid Term**	
18.	**3**	Architectural Design ▪ Architectural Design Decisions ▪ Architectural views ▪ Application Architectures	Pressman: Ch10 Sommerville: Ch6
19.	**3**	Architectural Patterns o Repository o Client Server o Layered	Sommerville: Ch6
20.	**3**	Behavioral Modelling ▪ Activity Diagrams	Ref. Material
21.	**3**	Interaction Models: ▪ Sequence Diagrams	Sommerville: Ch5
22.	**3**	Structured Analysis Modeling, Data Model: Entity Relationship Diagram (ERD).	Ref. Material
23.	**3**	Structural Models Class Diagrams	Sommerville: Ch5
24.		Case Studies of different software to practice UML Diagrams	Ref. Material
25.	**4**	Quality Management ▪ Software Quality (Definition and Concepts) ▪ Software Standards	Sommerville: Ch24

26.	**4**	Software Testing Black Box vs. White Box Testing Verification vs. Validation ▪ Unit Testing ▪ Component Testing ▪ System Testing	Sommerville: Ch8
27.	**5**	Software Project Management ▪ Introduction and concept ▪ Risk Management ▪ Team Work	Sommerville: Ch22
28.	**4**	Software Measurement ▪ Software Size Estimation	Sommerville: Ch24 Pressman: Ch33
29.		Software Measurement ▪ Software Pricing	Sommerville: Ch24
30.	**6**	Software Configuration Management Change and version management	Sommerville: Ch25
31.		**Final Term Exam**	

Students Outcomes (SOs)

S.#	**Description**
1	Apply knowledge of computing fundamentals, knowledge of a computing specialization, and mathematics, science, and domain knowledge appropriate for the computing specialization to the abstraction and conceptualization of computing models from defined problems and requirements Acquired an in-depth knowledge of software development processes and the role of design in the software development life-cycle.
2	Identify, formulate, research literature, and solve complex computing problems reaching substantiated conclusions using fundamental principles of mathematics, computing sciences, and relevant domain disciplines understand about systems modelling, analysis and design across both architectural and behavioral specifications.
3	Design and evaluate solutions for complex computing problems, and design and evaluate systems, components, or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations
4	Create, select, adapt and apply appropriate techniques, resources, and modern computing tools to complex computing activities, with an understanding of the limitations Students should have learned the Fundamental principles of formal specifications. Students should have clear understanding about software Management and Testing approaches.

Course Learning Outcomes (CLO)

Sr. #	**Unit #**	**Course Learning Outcomes**	**Blooms Taxonomy Learning Level**	**SOs**
		CLO's for Theory		
CLO-1	1	Explain the concept of software engineering along with its processes and deliverable	Remembering	1
CLO-2	1	Explain the concept of software engineering processes and deliverable	Understanding	1
CLO-3	2	Identify functional and non-functional requirements for a medium sized software system.	Analyzing	2
CLO-4	3	Construct appropriate design models for the structure and behavior of a medium sized software system.	Applying	2-4
CLO-5	4	Apply software testing and quality assurance techniques to medium sized software.	Applying	2,4
CLO-6	5,6	Demonstrate software project management skills and maintenance process.	Applying	2,4

CLO's for Lab				
CLO-7	3	Develop UML Diagrams and other associated documentation related to assigned project.	Creating	2,4

CLO Assessment Mechanism

Assessment Tool	CLO-1	CLO-2	CLO-3	CLO-4	CLO-5	CLO-6	CLO-7
Quizzes	Quiz1	Quiz1	Quiz2	Quiz3	Quiz4	Quiz4	
Assignments		Assignment 1	Assignment 2	Assignment 3 & Lab Assignment	Assign ment 4	Lab Assign ment	Lab Assign ment
Midterm Exam	Midterm Exam	Midterm Exam	Midterm Exam	Midterm Exam			
Final term Exam	Final term Exam		Final Term Exam	Final Term Exam			
Project	-	-	-	-	-	-	Lab Project

Policy & Procedures

- **Attendance Policy:** Every student must attend 80% of the lectures as well as laboratory in this course. The students falling short of required percentage of attendance of lectures/laboratory work, is not allowed to appear in the terminal examination.

- **Course Assessment:**

	Quizzes	Assignments	Mid Term Exam	Terminal Exam	Total
Theory(T)	15	10	25	50	100
Lab(L)		25	25	50	100

- **Grading Policy:** The minimum passing marks for each course is 50% (In case of LAB; in addition to theory, student is also required to obtain 50% marks in the lab to pass the course). The correspondence between letter grades, credit points, and percentage marks at CUI is as follows:

Grade	A	A-	B+	B	B-	C+	C	C-	D+	D	F
Marks	>= 85	80 - 84	75 – 79	71 - 74	68 – 70	64 – 67	61 - 63	58 - 60	54 - 57	50-53	< 50
Cr. Points	3.67-4:00	3.34-3.66	3.01-3.33	2.67-3.00	2.34-2.66	2.01-2.33	1.67-2.00	1.31-1.66	1.01-1.30	0.10-1.00	0.00

- **Missing Exam:** No makeup exam will be given for final exam under any circumstance. When a student misses the mid-term exam for a legitimate reason (such as medical emergencies), his grade for this exam will be determined based on the Department policy. Further, the student must provide an official excuse within one week of the missed exam.
- **Academic Integrity:** All CUI policies regarding ethics apply to this course. The students are advised to discuss their grievances/problems with their counsellors or course instructor in a respectful manner.

- **Plagiarism Policy:** Plagiarism, copying and any other dishonest behavior is prohibited by the rules and regulations of CUI. Violators will face serious consequences.