

<b>Course Title</b>	<b>Software Engineering</b>		
<b>Course Code</b>	<b>CC-212</b>		
<b>Credit Hours</b>	3 (3,0)		
<b>Category</b>	Computing Core		
<b>Prerequisite</b>	None		
<b>Co-Requisite</b>	None		
<b>Follow Up</b>	None		
<b>Course Learning Outcomes (CLOs)</b>	At the end of the course, the students will be able to:	<b>BT</b>	<b>PLO</b>
	CLO1: Describe various software engineering processes and activates	C1 (Describe)	1
	CLO2: Apply the system modeling techniques to model a medium size software system	C3 (Apply)	1,2
	CLO3: Apply software quality assurance and testing principles to medium size software systems	C3 (Apply)	2,3
	CLO4: Discuss key principles and common methods for software project management such as scheduling, size estimation, cost estimation, and risk analysis	C2 (Discuss)	1, 2
<b>Course Description</b>	<p>Introduction: Nature of Software, Overview of Software Engineering, Professional software development, Software engineering practice, Software process structure, Software Process Models: Waterfall Model, Incremental Model, Prototyping Model, Spiral Model, RAD Model. Agile Software Development: Agile process models, Agile development techniques. Introduction to Project Management, Introduction to Requirements Engineering, Functional and non-functional requirements. Analysis Model: Context models, Interaction models, Structural models, behavioral models, model driven engineering, Data modeling, Functional Modeling, Behavioral Modeling. Software Design: Data Design, Architectural Design, Component Level Design, User Interface Design. Object Oriented Analysis &amp; Design Basics: Introduction to UML, UML Diagrams. Use Case Modeling, Rational Rose overview, Use case modeling in Rational Rose. Domain Model: Identifying business classes, Domain Model Associations, Domain Model Attributes, Implementation of Sequence Diagram and Domain model in Rational Rose. Interaction Diagram: Sequence diagrams, Collaboration Diagrams, Implementation of Sequence and Collaboration diagrams in Rational Rose. Design Class Diagram, Mapping Design to Code. Software Testing Fundamentals. Design patterns, Software testing and quality assurance. Software evolution. Project Management: Project planning, configuration management. Software Process improvement.</p>		
<b>Text Book(s)</b>	1. Ian Sommerville, Software Engineering, 10th Edition, Pearson, 2015, ISBN-13: 978-0133943030.		
<b>Reference Material</b>	1. Carlo Ghezzi, Mehdi Jazayeri, Dino Mandrioli, Fundamentals of Software Engineering, 2nd Edition, Pearson, 2002, ISBN-13: 978-0133056990.		