Course Outline

<table>
<tr><td>Title</td><td>Software Engineering</td></tr>
<tr><td>Code</td><td>CS-212</td></tr>
<tr><td>Credit Hours</td><td>
Theory/week:
Weight: 3 Cr. Hrs.
Contact Hours: 3 Hrs.
Lectures: 2
Duration: 1.5 Hrs.
Lab/week:
Weight 0 Cr. Hrs.
Contact Hours 0 Hrs.
Labs. 0
Duration 0 Hrs.
</td></tr>
<tr><td>Prerequisite</td><td>Object Oriented Programming</td></tr>
<tr><td>Prerequisite Skill/Knowledge/Understanding</td><td>Understanding of programming concepts as well as knowledge of operating systems.</td></tr>
<tr><td>Required Study Hours</td><td>
<table>
<tr><th>TEACHING, LEARNING + ASSESSMENT ACTIVITIES</th><th>STUDY HOURS</th></tr>
<tr><td>32 x 1.5“ hr lectures</td><td>48</td></tr>
<tr><td>Lab activity / tutorials / presentations</td><td>NILL</td></tr>
<tr><td>Regular student’ Centered learning</td><td>35.5</td></tr>
<tr><td>Net Surfing</td><td>10</td></tr>
<tr><td>In’ course practical assignment(s)</td><td>NILL</td></tr>
<tr><td>In’ course writing assignment(s) 3 x 1.5 hrs</td><td>4.5</td></tr>
<tr><td>Preparation term examination + viva</td><td>24</td></tr>
<tr><td>Term examination + viva</td><td>8</td></tr>
<tr><td>Total</td><td>130</td></tr>
</table>
</td></tr>
<tr><td>Follow Up</td><td></td></tr>
<tr><td>Program Name</td><td>BS</td></tr>
<tr><td>Category</td><td>CS, SE, IT core</td></tr>
<tr><td>Aims and Objectives</td><td>
Aim: Application of software engineering practices to the development of software in information system development domain where professionalism, quality, schedule, and cost are important in producing an information system.
Objectives:

To understand the importance and need of information system software engineering
To discuss different software development models appropriate for the development and maintenance of software products
To introduce the basic project management concepts for the development of a high-quality product
To impart comprehensive knowledge regarding software development lifecycle
To demonstrate, with justification, an appropriate set of tools to support the development of a range of software projects
Application of object oriented concepts
Identify and analyze criteria and specifications appropriate to specific object oriented problems, and plan strategies for their solution

</td></tr>
<tr><td>Learning Outcomes</td><td>

The students will be able to develop information systems in a disciplined way, to meet their educational and professional requirements.
Students will develop the necessary documentation and follow the phase-by-phase approach towards software development.
The students will be proficient enough to analyze, evaluate and apply a set of CASE tools in a given area of software development in order to compete local and international market.
The execution of term project in order to implement the software engineering concepts.
Students will be able to select and apply appropriate Design Pattern

</td></tr>
</table>

<table>
<tr><td>Syllabus</td><td>The Scope of Software Engineering: Motivation and need, Importance and need of Sotware Engineering, Definition of Software Engineering, Introduction to software engineering vocabulary. Software Process Models: Waterfall Model, Incremental Model, Prototyping Model, Spiral Model, RAD Model. Introduction to Project Management: 4P's of Project Management. Introduction to Requirements Engineering. Analysis Model: Data modeling, Functional Modeling, Behavioral Modeling. Software Design: Data Design, Architectural Design, Component Level Design, User Interface Design. Object Oriented Analysis & Design Basics. Introduction to UML, UML Diagrams. Use Case Modeling, Rational Rose overview, Use case modeling in Rational Rose. Introduction of Interaction Diagrams, System Sequence Diagram; 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.</td></tr>
<tr><td>Text Book/s</td><td>A. Roger S. Pressman "Software Engineering- A practitioner's approach", 6th Ed.
B. Craig Larman "Applying UML and design Patterns", 2nd Ed.</td></tr>
<tr><td>Pedagogical Strategies</td><td>• Collaborative and cooperative learning
• Discussion groups and seminars
• Labs
• Demonstrations
• Ongoing reflection
• Self-evaluation</td></tr>
<tr><td>Reference Material</td><td>1. Ian Sommerville "Software Engineering", 6th Ed.
2. Fundamentals of Software Engineering By Carlo Ghezzi
3. Grady Booch "The Unified Modeling Language" 2nd Ed.
4. http://www.mhhe.com
5. http://www.sei-cmu.edu
6. www.rational.com</td></tr>
<tr><td>Instructional Aids/Resources</td><td>• Class facilities to Instructor and students for Lectures per Room
o 1 Network enabled PC
o 1 Multimedia
o 1 Whiteboard
o 1 Rostrum
o 1 Board marker/Month i.e. 4 Week Lectures (eight sessions)
o Max 45 students sitting facility
o CASE tools must be available in the lecture room as well as in labs.

• Evaluation and Support facilities
▪ Announced Testes i.e. pre mid and pre final Testes should not be conducted during class hours

• Photocopy facility for different Handouts
o Subject description document that includes student version of course outline, prerequisite test and its solution should be distributed in first class.
o Handout: - Required handouts as mentioned in lecture framework
o Quizzes and Solution: - Quizzes and their standard solution must be provided to students in handout form after each quiz.
o Case Studies and Tutorials: - Description of case study and handout of each tutorial should be provided.</td></tr>
<tr><td>Assessment Criteria</td><td>
<table>
<tr><th>Classroom Assessment</th><th>25%</th><th>Mid</th><th>35%</th><th>Final</th><th>40%</th><th>Total</th><th>100%</th></tr>
<tr><td>Quizzes and Test</td><td>10</td><td>Paper</td><td>35</td><td>Project</td><td>10</td><td></td><td></td></tr>
<tr><td>Assignment and Presentations</td><td>05</td><td></td><td></td><td>Paper</td><td>40</td><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>Recommendations</td><td>Marks division for sessional or project may vary on the basis of complexity of project or available time for project execution and documentation.</td></tr>
</table>

Lecture #	Topic/s	Source (Book-Chapter No. Section No.)	Required Study Hours	Recommendations for Learning Activities
1	The Scope of Software Engineering: Motivation and need for software engineering, Definition of Software Engineering, Introduction to software engineering vocabulary	A-1.2		
2	Software Process Models: Introduction, Linear Sequential Model, Prototyping Model	A-3.1, 3.2		
3	RAD Model, Evolutionary software process models, Incremental Model, Spiral Model	A-3.3, 3.4		
4	Introduction Project Management	A-21.1		
5	4 P's: People	A-21.2		
6	4 P's: Product	A-21.3		
7	4 P's: Process and Project	A-21.4, 21.5		
8	Introduction to Structured Analysis and Design	A-8.1		
9	Structured Analysis: Data Modeling (ERD and Data Dictionary)	A-8.3		
10	Structured Analysis: Functional Modeling (DFD, Process Specification, Mini Specification)	A-8.6		
11	DFD Case Study			
12	Structured Analysis: Behavioral Modeling (STD, Control Specification)	A-8.8		
13	Structured Design: Basic Concepts, Data Design and Introduction to Architectural Design	A-9.1, 9.2, 9.3		
14	Architectural Design and mapping of requirements to architectural design	A-9.4.2		
15	Component Level Design, User Interface Design	A-9.4.3, 9.4.4		
16	Pre Mid Review			
17	Introduction to UML, UML Diagrams	B-1		
18	Use Case Modeling	B-6		
19	System level use case Diagram, Use case modeling in Rational Rose	B-6		
20	Activity Diagram	B-9		
21	Introduction of Interaction Diagrams, System Sequence Diagram	B-15		
22	Domain Model: Identifying business classes, Domain Model Associations, Domain Model Attributes	B-11		
23	Implementation of Sequence Diagram, activity diagram and Domain model in Rational Rose	B-15		
24	Interaction Diagram: Sequence diagrams, Collaboration Diagrams	B-15		
25	Use case relationships, Analysis level use case Diagram	B-6		
26	Implementation of Sequence, Collaboration and analysis level use case diagrams in Rational Rose	B-15		
27	Introduction to Design Class Diagram	B-19		
28	Implementation of Design Class Diagram in Rational Rose			
29	Software Testing Fundamentals	A-14.1		
30	Testing Techniques	A-14.2, 14.3, 14.4, 14.5, 14.6		
31	Testing Strategy	A-13		

32	Pre Final Review			