

## **CSC-: Software Re-Engineering Syllabus**

### **General Information**

<b>Course Number</b>	CSC-
<b>Credit Hours</b>	3(Theory Credit Hour = 3)
<b>Prerequisite</b>	Software Construction and Development
<b>Course Coordinator</b>	Not Specified

### **Course Objectives**

This course is designed to give an introduction to Software Re-engineering. The main focus of this course is to present concepts and techniques necessary to effectively use system requirements captured in use cases to drive the development of a robust design model. In this course students will learn to apply UML 2.x notations including use case, architecture, objects, classes, components, subsystem, relationships and other UML diagrams. In modern software engineering processes are iterative thus, this course will also focus on an iterative approach towards design and analysis of software intensive systems.

The main topics include the terminology and the processes pertaining to software evolution, fundamental re-engineering techniques to modernize legacy systems including source code analysis, architecture recovery, and code restructuring, software refactoring strategies, migration to Object Oriented platforms, quality issues in re-engineering processes, migration to network-centric environments, and software integration, reverse engineering, program comprehension, source code transformation and refactoring strategies, software maintenance and re-engineering economics.

### **Catalog Description**

CSC-
------

## Course Content

Week No.	Topics	Suggested Readings (Chapters)
1	Introduction to Software Re-Engineering  Understanding the challenges of legacy project overview  Characteristics of legacy projects  Examples of legacy code and legacy infrastructure  Organizational factors that contribute to legacy projects  A plan for improvement  Legacy system types  Legacy migration methodology	1
2	Software Evolution  Evolution process  Evolution dynamics  Software maintenance  Legacy system management	-
3	Reverse Engineering & its techniques  Introduction to reverse engineering  The reverse engineering process  The tools  Reversing legality	

4	<p>Finding your starting point</p> <p>Fear and frustration</p> <p>Gathering data</p> <p>Code inspection</p>	
<b>First Mid Exams</b>		
6	<p>Preparing to refactor</p> <p>Forming team consensus</p> <p>Approval from Organization</p> <p>Decision for re factor or rewrite</p>	3
7	<p>Refactoring code</p> <p>Refactoring</p> <p>Legacy code trait</p> <p>Testing legacy code</p>	4
8&9	<p>Re-architecturing</p> <p>Re architecturing concepts</p> <p>Breaking monolithic application into modules</p> <p>Case study</p>	5
10	<p>The big rewrite process</p> <p>Project scope</p> <p>Learning from past</p> <p>What to do with database?</p>	6
<b>Second Mid Exams</b>		

12	Code restructuring	
13	Quality issues in re-engineering processes	
14	Tool support for Re-engineering, Challenges & Stakeholder aspiration	
15	Software maintenance and re-engineering economics	

### Text Book

1. Re-engineering legacy software, David Lorge Parnas, Chris Birchall

### Reference Material

--

### Course Learning Outcomes

	Course Learning Outcomes (CLO)
1	Explain the concepts and technique of software reengineering.
2	Apply reengineering techniques to maintain and modify software systems
3	Analyze and understand maintenance related problems associated with object oriented software
4	Able to perform complex design reengineering and reverse engineering problems

**CLO-SO Map**

	SO IDs											
CLO ID	GA1	GA2	GA3	GA4	GA5	GA6	GA7	GA8	GA9	GA10	GA11	GA12
CLO 1	1	0	0	0	0	0	0	0	0	0	0	0
CLO 2	0	1	0	0	0	0	0	0	0	0	0	0
CLO 3	0	0	1	0	0	0	0	0	0	0	0	0
CLO 4	0	0	1	0	0	0	0	0	0	0	0	0

**Approvals**

Originally Prepared By	Sehar Gul
Approved By	Not Specified
Last Update	04/03/2022