**Title of Subject : Software Re-Engineering (SW-415)**

**Discipline :** Software Engineering (7th Semester)

**Effective :** F16 Batch & onwards

**Pre-requisite :** Introduction to Software Engineering, Software requirement

 Engineering, Software Economics and Management, Software Design

 And Architecture

**Assessment :** Theory**:** 20% Sessional, 80% Written Semester Examination

## (20% Mid, 60% Final)

**Credit Hours :** 03 + 0 **Marks:** 100

 **Minimum Contact Hours:** 45

# Specific Objectives of course:

* This course explores the foundations of software maintenance by introducing several challenges linked to software evolution along with support tools to approach them. Also, the course covers various concepts related of software analysis and testing, along with practical tools, widely used in the open source community.
* This course Developers spend most of their time maintaining and updating existing software systems rather than designing new ones. Generally, evolving a software system, to ensure its continued integration and usefulness, is complex, human-intensive, costly, and an error-prone process.

**COURSE LEARNING OUTCOMES:**

Upon successful completion of the course, the student will be able to:

|  |  |  |  |
| --- | --- | --- | --- |
| **CLOs** | **Description** | **Taxonomy level** | **PLO** |
| 1 | Explain the concepts and technique of software re- engineering. | C1 | 1 |
| 2 | Apply reengineering techniques to maintain and modify software systems | C3 | 3 |
| 3 | Analyze and understand maintenance related problems associated with object oriented software systems. | C4 | 2, 3 |
| 4 | Able to perform complex design reengineering and reverse engineering problems. | C5 | 4 |

**PROGRAM LEARNING OUTCOMES (PLOs):**

The course is designed so that students will achieve the following PLOs:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | Engineering Knowledge: | ☑ | 7 | Environment and Sustainability:  | ☐ |
| 2 | Problem Analysis: | ☑ | 8 | Ethics: | ☐ |
| 3 | Design/Development of Solutions: | ☑ | 9 | Individual and Team Work: | ☐ |
| 4 | Investigation: | ☑ | 10 | Communication: | ☐ |
| 5 | Modern Tool Usage: | ☐ | 11 | Project Management: | ☐ |
| 6 | The Engineer and Society: | ☐ | 12 | Lifelong Learning: | ☐ |

**Course outline:**

* **SOFTWARE RE-ENGINEERING:**

The Reuse Process, Describing Reusable Components, Management Issues

* **FUNDAMENTAL RE-ENGINEERING TECHNIQUES:**

to modernize legacy systems including source code analysis, architecture recovery, and code restructuring, migration to Object Oriented platforms, migration to network-centric environments, and software integration,

* **SOFTWARE REFACTORING STRATEGIES:**

Forward and Reverse Engineering, program comprehension, source code transformation and refactoring strategies, software maintenance and re-engineering economics.

* **RE-ENGINEERING COST FACTORS:**

quality of the software, tool support, required data conversion, availability of expert staff for re-engineering

# Recommended Books:

1. Re-engineering legacy software, David Lorge Parnas, Chris Birchall, Safari Books, Shelter Island, NY, Latest Edition
2. Reengineering, Priyadarshi Tripathy and Kshirasagar Naik, John Wiley & Sons, Inc. Latest Edition
3. Software Maintenance and Evolution: a Roadmap, K.H.Bennett and V.T Rajlich, The Future of Software Engineering, ACM Press, Latest Edition

|  |  |
| --- | --- |
| **Approval:** |  |
| **Board of Studies:** | **Resolution No. 02** | **Dated: 29-08-2019** |
| **Board of Faculty:** | **Resolution No. 01** | **Dated: 07-10-2019** |
| **Academic Council:** | **Resolution No. 96.10** | **Dated: 07-10-2019** |