# COURSE CONTENT

| SOFTWARE ENGINEERING | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **V Semester: CSE (DS)** | | | | | | | | |
| **Course Code** | **Category** | **Hours / Week** | | | **Credits** | **Maximum Marks** | | |
| | | **L** | **T** | **P** | **C** | **CIA** | **SEE** | **Total** |
| **ACDD05** | **Core** | 3 | 0 | 0 | 3 | 40 | 60 | 100 |
| **Contact Classes: 48** | **Tutorial Classes: Nil** | **Practical Classes: Nil** | | | | **Total Classes:48** | | |
| **Prerequisites: There are no prerequisites for taking this course.** | | | | | | | | |

## I. COURSE OVERVIEW:

Software engineering is a discipline that allows us to apply engineering and computer science concepts in the development and maintenance of reliable, usable, and dependable software. This course is designed to present software engineering concepts and principles in parallel with the Software Development Life Cycle. At the end of this course, students will learn the project management for the purpose of delivering high-quality software that satisfies customer needs and is within the budget.

## II. COURSE OBJECTIVES:

**The students will try to learn:**

I. The principles of software engineering, software processes, and project management strategies including estimation, scheduling, and risk management.
II. The skills for effective requirements analysis, software design, and implementation**,** emphasizing modularity, user interface, and component-level design.
III. The software testing methodologies and project tracking techniques, covering quality assurance, debugging, and earned value analysis.

## III. COURSE OUTCOMES:

**At the end of the course, students should be able to**

**CO1** Illustrate process models, approaches and techniques for managing a software development process.
**CO2** Summarize the importance of project planning activities that accurately help in selection and initiation of individual projects and portfolios of projects in the enterprise.
**CO3** Explain software design model and behavior of a software system
**CO4** Develop the approaches for implementation, verification and validation including static analysis and reviews.
**CO5** Demonstrate the concept of risk management through risk identification, risk measurement and mitigation.
**CO6** Make use of earned value analysis and project metric for scheduling and improving the quality of software.

## IV. COURSE CONTENT:

**MODULE – I: SOFTWARE PROCESS AND PROJECT MANAGEMENT (10)**
Introduction to software engineering, software process, perspective, and specialized process models; Software project management: Estimation: LOC and FP based estimation, COCOMO model; Project scheduling: Scheduling, earned value analysis, risk management

**MODULE – II: REQUIREMENTS ANALYSIS AND SPECIFICATION (09)**
Software requirements: Functional and nonfunctional, user requirements, system requirements, software requirements document; Requirement engineering process: Feasibility studies, requirements elicitation and analysis, requirements validation, requirements management; Classical analysis: Structured system analysis, petri nets, data dictionary.

**MODULE – III: SOFTWARE DESIGN (09)**
Design process: Design concepts, design mode, design heuristic, architectural design architectural styles, architectural design, and architectural mapping using data flow.

User interface design: Interface analysis, interface design; Component level design: Designing class-based Components, traditional components

**MODULE – IV: TESTING AND IMPLEMENTATION (10)**
Software testing fundamentals: Internal and external views of testing, white box testing, basis path testing, control structure testing, black box testing, regression testing, MODULE testing, integration testing, validation testing, system testing and debugging; Software implementation techniques: Coding practices, refactoring.

**MODULE – V: PROJECT MANAGEMENT (10)**
Estimation: FP based, LOC based, make/buy decision; COCOMO II: Planning, project plan, planning process, RFP risk management, identification, projection; RMMM: Scheduling and tracking, relationship between people and effort, task set and network, scheduling; EVA: Process and project metrics.

## V. TEXTBOOKS:

1. Roger S. Pressman, "Software Engineering – A Practitioner's Approach", McGraw-Hill International Edition, 7th Edition, 2010.
2. Ian Somerville, "Software Engineering", Pearson Education Asia, 9th Edition, 2011.

## VI. REFERENCE BOOKS:

1. Rajib Mall, "Fundamentals of Software Engineering", PHI Learning Private Limited, 3rd Edition, 2009.
2. Pankaj Jalote, "Software Engineering, A Precise Approach", Wiley India, 1st Edition, 2010.

## VII. WEB REFERENCES:

1. http://www.softwareengineerinsider.com/articles/what-is-software-engineering.html
2. https://www.udacity.com/courses/software-engineering
3. http://www.tutorialspoint.com/software_engineering
4. http://computingcareers.acm.org/?page_id=12
5. http://en.wikibooks.org/wiki/Introduction_to_Software_Engineering

## VIII. MATERIALS ONLINE

1. Course template
2. Tutorial question bank
3. Tech talk topics
4. Open-ended experiments
5. Definitions and terminology
6. Assignments
7. Model question paper – I
8. Model question paper – II
9. Lecture notes
10. PowerPoint presentation
11. E-Learning Readiness Videos (ELRV)