COURSE CODE	COURSE TITLE	L	T	P	C
1151IT106	OBJECT ORIENTED SOFTWARE ENGINEERING	3	0	0	3

Course Category: Program Core

A. Preamble :

Software engineers are those who contribute by direct participation or by teaching, to the analysis, specification, design, development, certification, maintenance, and testing of software systems

B. Prerequisite Courses:

Sl No	Course Code	Course Name
1	1150CS201	Problem solving using C

C. Related Courses:

Sl No	Course Code	Course Name
1	1152IT131	Software Quality Assurance

D. Course Outcomes :

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

CO Nos.	Course Outcomes	Knowledge Level (Based on revised Bloom's Taxonomy)
CO1	Discuss about software development process models	K2
CO2	Identify the contemporary issues and discuss about coding standards	K2
CO3	Recognize the knowledge about testing methods and comparison of various testing techniques.	K2
CO4	Use the concept and standards of quality and getting knowledge about software quality assurance group.	K2

E. Correlation of COs with POs :

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	**M**											
CO2	**L**											
CO3	**M**			**M**								
CO4	**M**			**M**								

F. Course Content :

UNIT- I Introduction **L – 9**

Introduction to Software Engineering - Software Development process models – Agile Development - Project & Process - Project management - Process & Project metrics - Object Oriented concepts, Principles & Methodologies.

UNIT- II Planning & Scheduling **L – 9**

Software Requirements Specification, Software prototyping - Software project planning - Scope - Resources - Software Estimation - Empirical Estimation Models – Planning - Risk Management - Software Project Scheduling - Object Oriented Estimation & Scheduling.

UNIT -III Analysis **L – 9**

Analysis Modeling - Data Modeling - Functional Modeling & Information Flow - Behavioral Modeling-Structured Analysis - Object Oriented Analysis - Domain Analysis-Object oriented Analysis process - Object Relationship Model - Object Behaviour Model, Design modelling with UML.

UNIT -IV Design **L – 9**

Design Concepts & Principles - Design Process - Design Concepts - Modular Design - Design Effective Modularity - Introduction to Software Architecture - Data Design - Transform Mapping - Transaction Mapping - Object Oriented Design - System design process- Object design process - Design Patterns.

UNIT -V Implementation, Testing & Maintenance **L – 9**

Top - Down, Bottom-Up, object oriented product Implementation & Integration. Software Testing methods-White Box, Basis Path-Control Structure - Black Box - Unit Testing - Integration testing - Validation & System testing - Testing Tools – Software Maintenance & Reengineering.

TOTAL : 45 Periods

G. Learning Resources

i. Text Books :

1. Roger. S. Pressman and Bruce R. Maxim, “Software Engineering – A Practitioner’s Approach”, seventh Edition, McGraw Hill, 2015.
2. Ian Sommerville, “Software Engineering”, eighth edition, Pearson Education, New Delhi, 2011.
3. Bill Brykczynski, Richard D. Stutz ,”Software Engineering Project Management”, Wiley India Edition, IEEE computer society, 2007.
4. Craig Larman, Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd Edition), Pearson Education, 2008.

ii.Reference:

1. Fairley R, “Software Engineering Concepts”, second edition, Tata McGraw Hill,New Delhi, 2003.
2. Jalote P, “An Integrated Approach to Software Engineering”, third edition, Narosa Publishers, New Delhi, 2013.
3. Grady Booch, James Rumbaugh, Ivar Jacobson - "the Unified Modeling Language User Guide" - Addison Wesley, 1999.
4. Ali Bahrami, “Object Oriented Systems Development” 1st Edition, The McGraw-Hill Company, 1999.