SRM UNIVERSITY
(Under section 3 of UGC Act 1956)

FACULTY OF ENGINEERING AND TECHNOLOGY

CURRICULUM, PRE-REQUISITES/CO-REQUISITES CHART, AND SYLLABUS FOR B.TECH

UNDER CHOICE BASED FLEXIBLE CREDIT SYSTEM

REGULATION 2015

(For students admitted from 2015-16 onwards)

Specialization: B.Tech Software Engineering (Full Time)

Offering Department: Software Engineering

Approved in the 32nd Academic Council Meeting held on 23rd July 2016

CONTENTS

STUDENT OUTCOMES

The curriculum and syllabus for B.Tech programs (2015) conform to outcome based teaching learning process. In general, ELEVEN STUDENT OUTCOMES (a-k) have been identified and the curriculum and syllabus have been structured in such a way that each of the courses meets one or more of these outcomes. Student outcomes describe what students are expected to know and be able to do by the time of graduation. These relate to the skills, knowledge, and behaviors that students acquire as they progress through the program. Further each course in the program spells out clear instructional objectives which are mapped to the student outcomes.

The student outcomes are:

(a) an ability to apply knowledge of mathematics, science, and engineering

(b) an ability to design and conduct experiments, as well as to analyze and interpret data

(c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability

(d) an ability to function on multidisciplinary teams

(e) an ability to identify, formulate, and solve engineering problems

(f) an understanding of professional and ethical responsibility

(g) an ability to communicate effectively

(h) the broad education necessary to understand the impact of engineering solutions in global, economic, environmental, and societal context

(i) a recognition of the need for, and an ability to engage in life-long learning

(j) a knowledge of contemporary issues

(k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.

C-D-I-O INITIATIVE

The CDIO Initiative (CDIO is a trademarked initialism for **Conceive** — **Design — Implement — Operate**) is an innovative educational framework for producing the next generation of engineers. The framework provides students with an education stressing engineering fundamentals set in the context of Conceiving — Designing — Implementing — Operating real-world systems and products. Throughout the world, CDIO Initiative collaborators have adopted CDIO as the framework of their curricular planning and outcome-based assessment.

In the syllabus, every topic has been classified under one or more of C-D-I-O so that students and faculty alike are clear about the scope of learning to take place under each one of the topics.

SYMBOLS AND ABBREVIATIONS

B	--	**Courses under Basic Science and Mathematics**
BT	--	**Biotechnology Courses**
C-D-I-O	--	**Conceive-Design-Implement-Operate**
CE	--	**Civil Engineering Courses**
CS	--	**Computer Science and Engineering Courses**
CY	--	**Chemistry Courses**
Dept.	--	**Department of Software Engineering**
E with course code	--	**Elective Courses**
E	--	**Courses under Engineering Sciences**
EC	--	**Electronics and Communication Engineering Courses**
EE	--	**Electrical and Electronics Engineering Courses**
G	--	**Courses under Arts and Humanities**
IOs	--	**Instructional Objectives**
IT	--	**Information Technology Courses**
L	--	**Laboratory / Project / Industrial Training Courses**
LE	--	**Language Courses**
L-T-P-C	--	**L- Lecture Hours Per Week**
		T- Tutorial Hours Per Week
		P- Practical Hours Per Week
		C- Credits for a Course
M	--	**Courses with Multi Disciplinary Content**
MA	--	**Mathematics Courses**
ME	--	**Mechanical Engineering Courses**
NC	--	**NCC- National Cadet Corps**
NS	--	**NSS – National Service Scheme**
P	--	**Professional Core Courses**
PD	--	**Personality Development Courses**
PY	--	**Physics Courses**
SE	--	**Software Engineering Courses**
SO/SOs	--	**Student Outcomes (a-k)**
SP	--	**NSO- National Sports Organization**
YG	--	**Yoga Course**

FACULTY OF ENGINEERING AND TECHNOLOGY, SRM UNIVERSITY

DEPARTMENT OF SOFTWARE ENGINEERING

B.TECH SOFTWARE ENGINEERING CHOICE BASED FLEXIBLE CREDIT SYSTEM (CBFCS) Curriculum Under Regulations 2015 (For students admitted from 2015-16 onwards)

L Lecture Hours / Week | T Tutorial Hours / Week | C Credits | P Practical Hours / Week | L Laboratory Course | E Elective Courses | J Theory jointly with Lab | M Course with Multidisciplinary content

Category	Category - wise % of Credits	Year 1											Year 2										
		1st Semester						2nd Semester						1st Semester						2nd Semester				
		Course Code	Course Title	L	T	P	C	Course Code	Course Title	L	T	P	C	Course Code	Course Title	L	T	P	C	Course Code	Course Title	L	T	P	C
Arts & Humanities-G	8.33%	15LE101	English	2	0	0	2	15LE102	Value Education	2	0	0	2	15LE201E	German Language I	2	0	0	2	15LE207E	German Language II	2	0	0	2
		15PD101	Soft Skills I	1	1	0	1	15PD102	Soft Skills II	1	1	0	1	15LE202E	French Language I					15LE208E	French Language II				
								15NC101	NCC- National Cadet Corps	0	0	1	1	15LE203E	Japanese Language I					15LE209E	Japanese Language II				
								15NS101	NSS- National Service Scheme					15LE204E	Korean Language I					15LE210E	Korean Language II				
								15SP101	NSO- National Sports Organization					15LE205E	Chinese Language I					15LE211E	Chinese Language II				
								15YG101	Yoga					15PD201	Quantitative Aptitude & Logical Reasoning I	1	1	0	1	15PD202	Verbal Aptitude	1	1	0	1
	15		Total	3	1	0	3		Total	3	1	1	4		Total	3	1	0	3		Total	3	1	0	3
Basic Sciences - B	19.44%	15MA101	Calculus And Solid Geometry	3	1	0	4	15MA102	Advanced Calculus And Complex Analysis	3	1	0	4	15MA201	Transforms And Boundary Value Problems	4	0	0	4	15MA207	Probability And Queuing Theory	4	0	0	4
		15PY101	Physics	3	0	0	3	15PY102L	Material Science	2	0	2	3												
		15PY101L	Physics Laboratory	0	0	2	1	15CY102	Principles Of Environmental Science	2	0	0	2												
		15CY101	Chemistry	3	0	0	3																		
		15CY101L	Chemistry Laboratory	0	0	2	1																		
		15BT101	Biology For Engineers	2	0	0	2																		
	35		Total	11	1	4	14		Total	7	1	2	9		Total	4	0	0	4		Total	4	0	0	4
Engineering Sciences - E	8.33%	15CE101	Basic Civil Engineering	2	0	0	2	15ME101	Basic Mechanical Engineering	2	0	0	2												
		15EE101	Basic Electrical Engineering	2	0	0	2	15EC101	Basic Electronics Engineering	2	0	0	2												
		15ME105L	Engineering Graphics	1	0	4	3	15IT101L	Computer Hardware and Troubleshooting Laboratory	0	0	3	2												
		15CS101L	Programming Laboratory	1	0	2	2																		
	15		Total	6	0	6	9		Total	4	0	3	6		Total	0	0	0	0		Total	0	0	0	0
Professional - Core - P	42.22%							15IT102	Program Design And Development	3	0	0	3	15CS203	Computer System Architecture	3	0	0	3	15SE211	Requirements Engineering	3	0	0	3
								15IT102L	Program Design And Development Laboratory	0	0	2	1	15CS201J	Data Structures	3	0	2	4	15CS204J	Algorithm Design And Analysis	3	0	2	4
														15SE201J	Object Oriented Programming Using C++	3	0	2	4	15CS205J	Microprocessors And Microcontrollers	3	0	2	4
														15SE202	Software Engineering Principles	3	0	0	3	15SE203	Object Oriented Analysis And Design	2	2	0	3
														15CS202	Digital System Design	3	0	0	3	15SE204	Professional Ethics And Software Economics	2	0	0	2
	76		Total	0	0	0	0		Total	3	0	2	4		Total	15	0	4	17		Total	13	2	4	16
Prof - Electives - P	8.33%																								
	15		Total	0	0	0	0		Total	0	0	0	0		Total	0	0	0	0		Total	0	0	0	0
Project / Seminar / Internships - P	8.33%																								
	18		Total	0	0	0	0		Total	0	0	0	0		Total	0	0	0	0		Total	0	0	0	0
Open Electives	-																								
	6		Total	0	0	0	0		Total	0	0	0	0		Total	0	0	0	0		Total	0	0	0	0
Total	180			20	2	10	26			17	2	8	23			22	1	4	24			20	3	4	23
			Contact hours	32					Contact hours	27					Total contact hours	27					Total Contact hours	27			

FACULTY OF ENGINEERING AND TECHNOLOGY, SRM UNIVERSITY

DEPARTMENT OF SOFTWARE ENGINEERING

B.TECH SOFTWARE ENGINEERING CHOICE BASED FLEXIBLE CREDIT SYSTEM (CBFCS) Curriculum Under Regulations 2015 (For students admitted from 2015-16 onwards

/ Week	T	Tutorial Hours / Week		C Credits	P Practical Hours / Week	L	Laboratory Course			E Elective Courses	J	Theory jointly with Lab			M Cou
Year 3										Year 4					
1st Semester					2nd Semester					1st Semester					
	L	T	P	C	Course Code	Course Title	L	T	P	C	Course Code	Course Title	L	T	P	C	Course Code	Co
on & Reasoning	1	1	0	1	15PD302	Quantitative Aptitude & Logical Reasoning II	1	1	0	1								
	1	1	0	1		Total	1	1	0	1		Total	0	0	0	0		Total
thematics	4	0	0	4														
	4	0	0	4		Total	0	0	0	0		Total	0	0	0	0		Total
	0	0	0	0		Total	0	0	0	0		Total	0	0	0	0		Total
ystems	3	0	2	4	15SE312	Software Modeling And Analysis	3	0	0	3	15IT304J	Web Programming	3	0	2	4		
g In JAVA	3	0	2	4	15SE313	Software Project Management	3	0	0	3	15SE411	Security In Networks And Software Development	3	0	0	3		
chitecture And	3	0	0	3	15SE314J	Software Verification And Validation	3	0	2	4	15SE401M	Multi Disciplinary Design	3	0	0	3		
nagement Systems	3	0	2	4	15IT303J	Computer Networks	3	0	2	4								
	12	0	6	15		Total	12	0	4	14		Total	9	0	2	10		Total
-I				3		Dept Elective-II				3		Dept Elective-IV				3		
						Dept Elective-III				3		Dept Elective-V				3		
	0	0	0	3		Total	0	0	0	6		Total	0	0	0	6		Total
ining (To be done ter)	0	0	3	2	15SE375L / 15SE380L / 15SE385L /	Minor Project I / Seminar I / MOOCs I / Industrial Module I	0	0	3	2	15SE376L / 15SE381L / 15SE386L /	Minor Project II / Seminar II / MOOCs II / Industrial Module I	0	0	3	2	15SE496L	Major Project

B.Tech Software Engineering (Regulations 2015)					
Department Elective I					
E	Course Title	L	T	P	C
1	Theory Of Computation	3	0	0	3
)E	Computational Logic	3	0	0	3
1E	Neuro Fuzzy And Genetic Programming	3	0	0	3
1E	Principles Of Programming Languages	3	0	0	3
1E	Python Programming	2	2	0	3
1E	Linux Internals	3	0	0	3
1E	E-Commerce	3	0	0	3
5E	Design Patterns	3	0	0	3
)E	Human Computer Interface	3	0	0	3
4E	Multimedia Systems	3	0	0	3
	Department Elective II & III				
5E	System Software	3	0	0	3
5E	Principles Of Compiler Design	3	0	0	3
7E	Distributed Operating System	3	0	0	3
5E	Digital Image Processing	3	0	0	3
8E	Programming In PHP	2	2	0	3
)E	Visual Programming	2	2	0	3
4E	Advanced Java Programming	2	2	0	3
1E	Game Programming	2	2	0	3
1E	Mobile Application Development	2	2	0	3
1E	Enterprise Resource Planning	3	0	0	3
)E	Database Administration	2	2	0	3
1E	Data Mining And Analytics	3	0	0	3
1E	Agile Software Process	3	0	0	3
)E	Xml And Webservices	3	0	0	3
5E	Pervasive Computing	3	0	0	3
5E	Embedded System Design	3	0	0	3
7E	Cryptography	3	0	0	3
1E	Parallel Programming Using OpenCL	3	0	0	3

B.Tech Software Engineering (Regulations 2015)		
Department Elective IV & V		
COURSE CODE	Course Title	L
15CS401	Artifical Intelligence	3
15CS421E	Natural Language Processing	3
15CS422E	Knowledge Based Decision Support Systems	3
15CS424E	Semantic Web	3
15IT422E	Internet Of Things	2
15IT423E	Data Science And Big Data Analytics	2
15IT424E	Business Intelligence And Analytics	3
15SE421E	Software Reliability	3
15SE422E	Software Quality Assurance	3
15SE423E	Software Configuration Management	3
15SE424E	Analysis Of Software Artifacts	3
15SE425E	Software Maintenance And Administration	3
15CS332E	Wireless Sensor Networks	3
15CS334E	Network Programming	3
15CS433E	Network Design And Management	3
15CS434E	Network Security	3
15CS423E	Software Defined Networks	3
15CS425E	Service Oriented Architecture	3
15SE426E	Software Measurements And Metrics	3
15SE427E	Wireless And Mobile Communication	3

Courses Customised to Other Departments

COURSE CODE	Course Title	L
15SE251E	Principles of Object Oriented Programming	3

Regarding open electives, all core / elective courses ca
listed every semester, under open electives,
based on the availability of resources and demand

SRM UNIVERSITY

DEPARTMENT OF SOFTWARE ENGINEERING

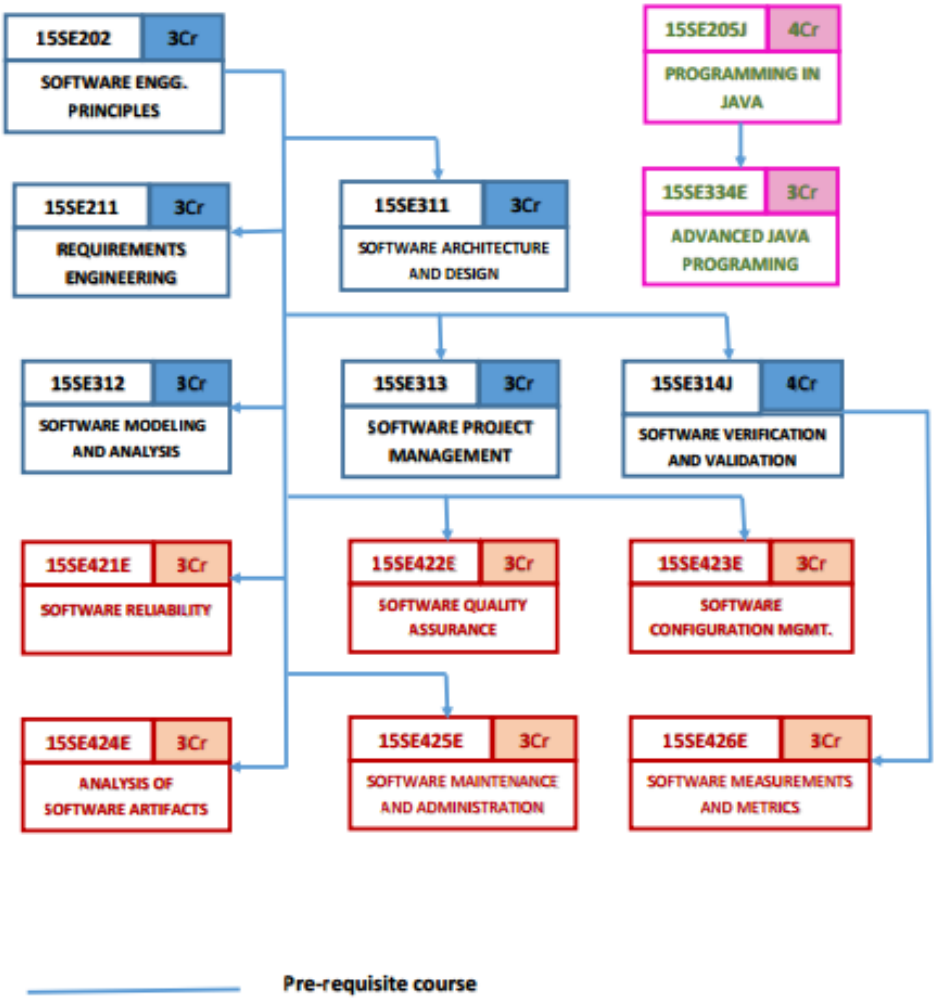
PRE/CO REQUISITES LIST

Course code	Course title	Pre-requisite	Co-requisite
15SE211	Requirements Engineering	15SE202	NIL
15SE311	Software Architecture And Design	15SE202	NIL
15SE312	Software Modeling And Analysis	15SE202	NIL
15SE313	Software Project Management	15SE202	NIL
15SE314J	Software Verification and Validation	15SE202	NIL
15SE334E	Advanced Java Programming	15SE205J	NIL
15SE421E	Software Reliability	15SE202	NIL
15SE422E	Software Quality Assurance	15SE202	NIL
15SE423E	Software Configuration Management	15SE202	NIL
15SE424E	Analysis of Software Artifacts	15SE202	NIL
15SE425E	Software Maintenance and Administration	15SE202	NIL
15SE426E	Software Measurements and Metrics	15SE314J	NIL

SRM UNIVERSITY

DEPARTMENT OF SOFTWARE ENGINEERING

Pre-requisite and Co-requisite



15SE201J	Object Oriented Programming Using C++	L	T	P	C
		3	0	2	4
Co-requisite:	NIL				
Prerequisite:	NIL				
Data Book / Codes/Standards	NIL				
Course Category	P PROFESSIONAL CORE				
Course designed by	Department of Software Engineering				
Approval	32nd Academic Council Meeting, 23rd July, 2016				

PURPOSE	Real world is full of objects and problems can be best solved using object oriented approach. The pioneer programming language to implement object oriented features is C++. This course aims at building object oriented skills through programming in C++.

INSTRUCTIONAL OBJECTIVES		**STUDENT OUTCOMES**						
At the end of the course, student will be able to								
1.	Apply the basic object oriented features	a	c					
2.	Develop Generic programming skills	a	c					
3.	Apply appropriate data structures and solve complex problems by applying the skills acquired so far	a	b	i				
4.	Design problem into classes and develop a full working code	a	c					
5.	Develop programs using Streams, files, templates and handle exceptions	a	c	i				

Session	Description of Topic	Contact hours	C-D-I-O	IOs	Reference
	UNIT I: INTRODUCTION TO OBJECT-ORIENTED PROGRAMMING	**10**			
1.	Need of Object-Oriented Programming - Comparison of procedural programming and Object-Oriented Programming	1	C	1	1
2.	Characteristics of Object-Oriented Languages	1	C	1	1
3.	C++ Programming Basics: Basic Program Construction	1	C	1	1
4.	Data Types, Variables, Constants	1	C	1	1
5.	Type Conversion, Operators, Library Functions	1	C	1	1
6.	Loops and Decisions, Structures	2	C	1,2	1
7.	Functions – Simple Functions, Passing arguments, Returning values, Reference Arguments	1	C	1,2	1
8.	Recursion, Inline Functions, Default Arguments Storage Classes	1	C	1,2	1
9.	Arrays – Strings	1	C	1,2	1
	UNIT II: FEATURES OF OBJECT-ORIENTED PROGRAMMING	**11**			
10.	Introduction to Classes and Objects	1	C	1	1,2,3
11.	Constructors and its types, Destructors	1	C,I	1,2	1,2,3
12.	Passing Objects as Function arguments and Returning Objects from Functions	1	C,I	1,3,4	1,2,3
13.	Operator Overloading	1	C,I	1,2	1,2,3
14.	Inheritance	2	C,I	1,3,4	1,2,3
15.	Overloading Member Functions	1	C,I	1,2	1,2,3
16.	Pointers	2	C,I	1,3,4	1,2,3
17.	Virtual Functions – Friend Functions, Static Functions	2	C,I	1,2	1,2
	UNIT III: STREAMS AND FILES	**7**			
18.	Streams: Classes and Errors	1	C	5	1,3,4
19.	Disk File I/O with Streams	1	C,I	5	1,3,4
20.	Files: File Pointers, Error handling in File I/O, File I/O with member Functions	3	C,I	3,5	1,3,4
21.	Overloading the extraction and Insertion Operators	1	C,I	5	1,3,4

22.	Multi File Programs	1	C,I	5	1,3,4
	UNIT IV: TEMPLATES, EXCEPTIONS	**7**			
23.	Templates – Function templates, Class templates	2	C	5	1,3,4
24.	Exceptions: Need of Exceptions, keywords, Simple and Multiple Exceptions	3	C,I	5	1,3,4
25.	Re-throwing Exception and Exception Specifications, Custom Exception	2	C,I	5	1,3,4
	UNIT V: STANDARD TEMPLATE LIBRARY	**10**			
26.	Introduction to STL: Containers, Algorithms, iterators - potential problems with STL	2	C,I	5	1
27.	Algorithms: find(), count(), sort(),search(),merge()	1	C,I	5	1
28.	Function Objects: for_each() , transform()	1	C,I	5	1
29.	Sequence Containers: vectors, Lists, Deques	2	C,,I	3,5	1
30.	Iterators and specialized iterators	1	C,I	5	1
31.	Associative Containers: Sets and Multisets Maps and multimaps	2	C,I	3,5	1
32.	Storing User-Defined Objects – Function Objects	1	C	5	1
	Total contact hours	**45**			

LEARNING RESOURCES	
Sl.No.	**TEXT BOOKS**
1.	Robert Lafore, "Object-Oriented Programming in C++", 4th edition, SAMS Publishing, 2008
2.	Deitel, "C++ How to Program", 6th edition, PHI publication, 2008
3.	R. Subburaj, "Object Oriented Programming With C++ ", Vikas Publishing House, New Delhi, Revised Edition, 2013.
	REFERENCE BOOKS/OTHER READING MATERIAL
4.	E.Balaguruswamy "Object Oriented Programming with C++", 6th edition, Tata McGraw Hill Education, 2015
5.	Joyce Farrell, "Object Oriented Programming", 4th edition, Cengage learning, 2009
6.	Nicolai M. Jossutis, "Object-Oriented Programming in C++", Wiley Publications, 2002
7.	BjarneStroustrup ,"The C++ Programming Language", 4th Edition, Addison Wesley, 2015
8.	Stanley Lippman, JoseeLajoie, Barbara E. Moo ,"C++ Primer", 5th Edition, Addison Wesley, 2015

S. No.	**Description of Experiments**	**Contact hours**	**C-D-I-O**	**IOs**	**Reference**
Each student is assigned with an application in Session 1. **Students have to complete the below listed experiments with respect to the assigned application.**					
1.	Identifying appropriate data types, variables and simple programs to understand the basic program structure	2	C,D,I	1	1-8
2.	Programs for control structures and loops	2	C,D,I	1	1-8
3.	Simple Programs to construct a class structure with methods and arguments	2	C,D,I	1,2	1-8
4.	Programs to develop their skills on Inheritance	2	C,D,I	1,4	1-8
5.	Programs to improve their skills on polymorphism	2	C,D,I	1,4	1-8
6.	Programs to construct Functions, Inline functions, and Virtual functions	4	C,D,I	1,4	1-8
7.	Develop a complete logic for the assigned application including all the concepts done so far	4	C,D,I	3,4	1-8
8.	Programs to improve the skills on reading and storing from and to files	2	C,D,I	5	1-8
9.	Programs for manipulating pointers	4	C,D,I	1,2,4	1-8
10.	Programs to construct templates and handle exceptions	2	C,D,I	5	1-8
11.	Programs to construct a STL for Sequential containers and iterators	2	C,D,I	5	1-8
12.	Programs to construct a STL for Associative containers	2	C,D,I	5	1-8
	Total Contact Hours	**30**			

Course nature				**Theory + Practical**			
Assessment Method – Theory Component (Weightage 50%)							
In-semester	**Assessment tool**	Cycle test I	Cycle test II	Cycle Test III	Surprise Test	Quiz	**Total**
	Weightage	**10%**	**15%**	**15%**	**5%**	**5%**	**50%**
						End semester examination Weightage :	**50%**
Assessment Method – Practical Component (Weightage 50%)							
In-semester	**Assessment tool**	Experiments	Record	MCQ/Quiz/Viva Voce	Model examination		**Total**
	Weightage	**40%**	**5%**	**5%**	**10%**		**60%**
						End semester examination Weightage :	**40%**

15SE202	Software Engineering Principles	L	T	P	C
		3	0	0	3
Co-requisite:	NIL				
Prerequisite:	NIL				
Data Book / Codes/Standards	NIL				
Course Category	P PROFESSIONAL CORE				
Course designed by	Department of Software Engineering				
Approval	32nd Academic Council Meeting, 23rd July, 2016				

PURPOSE	The main purpose of this course is to impart knowledge on the basic principles of software engineering							
INSTRUCTIONAL OBJECTIVES		**STUDENT OUTCOMES**						
At the end of the course, student will be able to								
1	Understand the software life cycle models and software development process	a						
2	Elicit, analyze and specify software requirements through a productive working Relationship with project stakeholders	a	g					
3	Understand the importance of modeling and modeling languages	c	g					
4	Develop correct and robust software products	d	h					
5	Adaptation of Software maintenance and emerging trends in software engineering	d	j					

Session	Description of Topic	Contact hours	C-D-I-O	IOs	Reference
	UNIT I: INTRODUCTION	**9**			
1	Software Engineering-Software Process- Generic process model	1	C	1	1
2	Process models	4	C	1	1
3	Agile development-Agile Process- Extreme Programming	2	C	1	1
4	Other Agile process models : Adaptive process models ,Scrum,Dynamic Systems Development Method and Crystal	2	C	1	1
	UNIT II: UNDERSTANDING REQUIREMENTS	**9**			
5	Principles that guide practice –Understanding requirements	3	C	1,2	1,3
6	Requirement Modeling : Scenarios, Information and Analysis classes	3	C,D	1,2,3	1,3
7	Requirements Analysis-Requirements Modeling Strategies	3	C	2,3,4	1,3
	UNIT III: DESIGN	**9**			
8	The design process - Design concepts	2	C,D	3	1
9	The design model –Architectural design-Software architecture	1	C	3	1
10	Component level design-user interface design	2	C,D	3	1
11	User Interface Design	2	C,D	3	1
12	Pattern oriented design-Web application design	2	C,D	3	1

	UNIT IV: SOFTWARE IMPLEMENTATION AND TESTING	**9**			
13	Modern Programming Language Features – Implementation Issues and Solutions	1	C	4	4
14	Structured coding Techniques-Coding Styles-Standards and Guidelines- Documentation Guidelines	2	C	4	4 ,2
15	Software Testing strategies on Conventional software ,Object oriented software and Web applications	3	C,D	4,5	1
16	Validation Testing- System Testing- Art of Debugging.	3	C,D	4,5	1
	UNIT V: SOFTWARE MAINTANENCE AND EMERGING TRENDS IN SOFTWARE ENGINEERING	**9**			
17	Software maintenance and Reengineering	4	C	4,5	1
18	Latest Trends: Technology evolution -Identifying Soft trends	2	C	4,5	1
19	Technology directions-Tools related trends	3	C	4,5	1
	Total contact hours	**45**			

LEARNING RESOURCES	
Sl. No.	**TEXT BOOKS**
1	Roger S Pressman, "Software Engineering – A Practitioner's Approach", 7th edition, Tata McGraw Hill Education, 2014.
2	Ian Somerville "Software Engineering", 9th edition, Pearson Education, 2010.
	REFERENCE BOOKS/OTHER READING MATERIAL
3	Hans Van Vliet, "Software Engineering: Principles and Practices", Wiley 2008.
4	Richard Fairley, "Software Engineering Concepts", Tata McGraw Hill Education, 2008.

Course nature					**Theory**		
Assessment Method (Weightage 100%)							
In-semester	**Assessment tool**	**Cycle test I**	**Cycle test II**	**Cycle Test III**	**Surprise Test**	**Quiz**	**Total**
	Weightage	**10%**	**15%**	**15%**	**5%**	**5%**	**50%**
					End semester examination Weightage :		**50%**

15SE203	Object Oriented Analysis And Design	L	T	P	C
		2	2	0	3
Co-requisite:	NIL				
Prerequisite:	NIL				
Data Book / Codes/Standards	NIL				
Course Category	P PROFESSIONAL CORE				
Course designed by	Department of Software Engineering				
Approval	32nd Academic Council Meeting , 23rd July 2016				

PURPOSE	The purpose of the course is to practice popular technical approach for analyzing, designing an application, system, or business by applying the object-oriented paradigm and visual modeling throughout the development life cycles to foster better stakeholder communication and product quality.

INSTRUCTIONAL OBJECTIVES		**STUDENT OUTCOMES**						
At the end of the course, student will be able to								
1.	Understand the basics object model for System development.	j						
2.	Apply the various modeling techniques using UML approach.	b						
3.	About building high quality system for different real world issues.	c						

Session	Description of Topic	Contact hours	C-D-I-O	IOs	Reference
	UNIT I: INTRODUCTION	**6**			
1.	Complexity in Traditional Systems	1	C	1	1,3,4
2.	The Object Model	2	C	1	1,3,4
3.	Classes and Objects	2	C	1	1,3,4
4.	Classification	1	C	1	1,3,4
	UNIT II: STATIC MODELING	**6**			
5.	What is UML?	1	C	2	2,3,4
6.	Use Case Diagram	2	C,D	2,3	2,3,4
7.	Domain Models	1	C,D	2,3	2,3,4
8.	UML Class Diagram	2	C,D	2,3	1,3,4
	UNIT III: DYNAMIC MODELING	**6**			
9.	Interaction and Package Diagram	2	C,D	2,3	2,3,4
10.	Activity Diagrams and Modeling	1	C,D	2,3	1,3,4
11.	State Machine Diagram and Modeling	1	C,D	2,3	1,3,4
12.	UML Component Diagram	1	C,D	2,3	1,3,4
13.	UML Deployment Diagram	2	C,D	2,3	1,3,4
	UNIT IV: GoF DESIGN PATTERNS	**6**			
14.	Object Design	1	C	1	2
15.	What are Patterns?	1	C	1	2
16.	Applying GoF Design Patterns	2	C,D	2	2
17.	Design Persistence Framework	2	C	2,3	2
	UNIT V: APPLICATIONS	**6**			
18.	Satellite Based Navigation	1	D,I	2,3	1
19.	Traffic Management	1	D,I	2,3	1
20.	Crypt Analysis	2	D,I	2,3	1
21.	Weather Monitoring Station	1	D,I	2,3	1
22.	Vacation Tracking System	1	D,I	2,3	1
	Total contact hours	**30**			

Sl. No.	Description of Tutorials	Contact hours	C-D-I-O	IOs	Reference
1	Implementation of Use Case Diagram	3	C,D,I	1,2,3	2,3,4
2	Implementation of Class Diagram	4	C,D,I	2,3	1,3,4
3	Implementation of Interactions Diagram	2	C,D,I	2,3	2,3,4
4	Implementation of State Machine Diagram	2	C,D,I	2,3	1,3,4
5	Implementation of Activity Diagram	2	C,D,I	2,3	1,3,4
6	Implementation of Component Diagram	3	C,D,I	2,3	1,3,4
7	Implementation of Deployment Diagram	2	C,D,I	2,3	1,3,4
8	Generate Skeleton code of Class Diagram for your Mini-Project	4	D,I	1,2,3	3
9	Implement the generated code using C++/Java Applications	4	C,D,I	1,2,3	3
10	Perform Test Case and Test Plan for implemented Mini-Project	4	C,D	3	4
	Total contact hours	30			

Note: Each Student is assigned with a mini project for which implementation to be carried out

LEARNING RESOURCES	
Sl. No.	**TEXT BOOKS**
1.	Grady Booch, Robert A. Maksimchuk, Michael W. Engle,"Object-Oriented Analysis and Design with Applications" ,Addison-Wesley Professional; 3 edition (April 30, 2007)ISBN-13**:** 978-0201895513
2.	Craig Larmen, "Applying UML and Patterns", Prentice Hall; 3 edition (October 30, 2004)ISBN-13**:** 978-0131489066
	REFERENCE BOOKS/OTHER READING MATERIAL
3.	Brett McLaughlin,"Head First Object-Oriented Analysis and Design",O'Reilly Media; 1 edition (December 7, 2006)ISBN-13**:** 978-0596008673
4.	Ali Bahrami , "Object Oriented Systems Development", McGraw Hill Eduction, Indian Edition, 2004, ISBN-13:978-0-07-026512-7

Course nature				**Theory**	
Assessment Method – Theory Component (Weightage 50%)					
In-semester	**Assessment tool**	Cycle test I	Model Exam	Tutorial Project	**Total**
	Weightage	**10%**	**15%**	**25%**	**50%**
				End semester examination Weightage :	**50%**

15SE204	Professional Ethics And Software Economics	L	T	P	C
		2	0	0	2
Co-requisite:	NIL				
Prerequisite:	NIL				
Data Book / Codes/Standards	Clark's Table, IS : 456-2000				
Course Category	P PROFESSIONAL CORE				
Course designed by	Department of Software Engineering				
Approval	32nd Academic Council Meeting, 23rd July, 2016				

PURPOSE	To exposure the aspects of professional ethics and software economics as an integral component of the undergraduate curriculum							
INSTRUCTIONAL OBJECTIVES		**STUDENT OUTCOMES**						
At the end of the course, student will be able to								
1.	Gain knowledge, skills, and attitudes that software engineers must possess to practice software engineering professionally, responsibly, and ethically.	a	d	e	f			
2.	Develop professionals with idealistic, practical and moral values	c	d	f				
3.	Comprehend the role of professional societies and software engineering standards	a	d	f	h			
4.	Comprehend the basic fundamentals of software economics	c	h					
5.	*Comprehend* and be able to apply the key software engineering economic fundamentals to real-world software economic issues	c	h					

Session	Description of Topic	Contact hours	C-D-I-O	IOs	Reference
	UNIT 1- GROUP DYNAMICS AND PYSCHOLOGY	**7**			
1.	Dynamics of working in teams and groups : Introduction to teams, Methods of Assessing and Evaluating Team Functioning	1	C	1	1
2.	Dealing with multicultural environments: Understanding Culture and Diversity-Understanding Cohesion and Collaboration	2	C	1	1
3.	Individual cognition : Improving Creativity and Innovation, Cognitive problem complexity	1	C	1	1
4.	Interacting with stakeholders-Dealing with uncertainty and ambiguity-Improving Problem Solving and Decision Making-presentation skills	3	C	1	1,4
	UNIT II- PROFESSIONALISM	**7**			
5.	Accreditation-certification-licensing	1	C	2	5
6.	Professionalism and Codes of ethics-Importance of Codes, Abuse of Codes, Limitations of Codes, Ethical Relativism, Justification of Codes	2	C	2	2
7.	Professional conduct: Professional Rights – Employee Rights.	1	C	2	2
8.	Professional Concerns: Introduction, Environmental Ethics, Computer Ethics	3	C	2	2
	UNIT III- ROLE OF PROFESSIONAL SOCIETIES AND SOFTWARE ENGINEERING STANDARDS	**6**			
9.	Nature and role of professional societies : Professional responsibilities	1	C	3	6
10.	Confidentiality and Proprietary Information,	1	C	3	6
11.	Conflict of Interest, Competitive bidding, whistle-blowing	2	C	3	6
12.	Nature and role of software engineering standard: Engineering standard, the standard of care, design standard, range of standard of practice.	2	C	3	3

	UNIT IV- SOFTWARE ECONOMICS	**6**			
13.	Introduction-Emphasis on Software Economics-Current status of software Economics	1	C	4	7
14.	Software Economics Road Map-Link Between software economics and policy,Software R&D Investment Policy Framework-Monitoring & Control for Dynamic Investment	2	C	4	7
15.	Improving Software Economics within an Enterprise-Modelling Costs, Benefits, and Value- Impact of Software Economics	2	C	4	7
16.	Employment Contracts-Legal Issues	1	C	4	7
	UNIT V- TRENDS IN SOFTWARE ECONOMICS	**4**			
17.	Case Study 1- Impact of Properly Licensed Software	2	C,O	5	8
18.	Case study 2: Current Growth and Impact of Software on Indian Economy	2	C,O	5	9
	Total Contact Hours	**30**			

LEARNING RESOURCES	
Sl. No.	**TEXT BOOKS**
1.	Timothy M. Franz by "*Group Dynamics and Team Interventions: Understanding and Improving Team Performance*", Wiley Black-Well, April 2012
2.	Mike W.Martin, Ronald Schinzinger, "*Introduction to Engineering Ethics* "2nd,Edition, Tata McGraw-Hill Higher Education,2009
3.	James .W. Moore ,"*Road Map to Software Engineering-Standards based Guide*" Wiley-IEEE Computer Society Press,2006.
	REFERENCE BOOKS/OTHER READING MATERIAL
4	Rio de Janeiro, "*Stake Holder Relationship Management*", 7th Project Management National Benchmarking Forum ,PMI Chapter, Nov. 2010**.**
5	Pierre Bourque, Richard E. Fairley," SWEBOK V3.0 *Guide to the Software Engineering Body of Knowledg*,IEEE Computer Society, January 2014
6	Charles D Fleddermann, "*Engineering Ethics", Prentice Hall, New Mexico*, 1999
7	Barry W. Boehm, Kevin J. *Sullivan "Software Economics: RoadMap*",1981.
8	INSEAD, Business School of the World, *"Competitive Advantage: Impact of Properly Licensed Software",* 2013.
9	Dr. Rajesh Kumar Jhamb,"*Contribution of Software Industry in the Growth of Indian Economy in the Last Decade*", ZENITH International Journal of Business Economics & Management Research Vol.1 Issue 3, December 2011,

Course nature				**Theory**			
Assessment Method (Weightage 100%)							
In-semester	**Assessment tool**	Cycle test I	Cycle test II	Cycle Test III	Surprise Test	Quiz	**Total**
	Weightage	**10%**	**15%**	**15%**	**5%**	**5%**	**50%**
					End semester examination Weightage :		**50%**

15SE211	Requirements Engineering	L	T	P	C
		3	0	0	3
Co-requisite:	NIL				
Prerequisite:	15SE202				
Data Book / Codes/Standards	NIL				
Course Category	P PROFESSIONAL CORE				
Course designed by	Department of Software Engineering				
Approval	32nd Academic Council Meeting, 23rd July, 2016				

PURPOSE	Aids the reader to systematically establish, define and manage the requirements for a large, complex, changing and software-intensive systems, from technical, organizational and management perspectives

INSTRUCTIONAL OBJECTIVES		**STUDENT OUTCOMES**						
At the end of the course, student will be able to								
1	Identify the Sources and collect, organize and classify the requirements	a	e	g				
2	Improve their ability, model requirements for any Systems	a	c					
3	Categorize functional and non-functional requirements	a						
4	Familiarize with the various requirements elicitation and validation techniques	a						
5	Expose to the latest requirement Engineering tools	a	e	k				

Session	Description of Topic	Contact hours	C-D-I-O	IOs	Reference
	UNIT I: INTRODUCTION	**10**			
1	Introduction to requirements, Requirement Engineering	1	C	1	1
2	Basics Definitions, V-model of requirements, requirement types	1	C,D	1	1
3	Classification of Requirements, Requirement gathering relevant to Software life cycle models	1	C,D	1,2,3	1,3
4	Requirement Engineering Process Basic Framework , SPIRAL, RAD, Requirement Engineering Process Maturity Model	1	C	1,3	1
5	Generic Process for Requirement Engineering, Levels of requirement	1	C,D	1	1
6	System Model for Requirement Engineering	1	C	1	2,5
7	Representation of Requirement – Data flow, ER Diagram, View point , Controlled Requirement Expression, Structured Analysis and Design Technique, Viewpoint Oriented Requirements Definition	2	C	1,2	1,2,5
8	Object Oriented Methods of Requirement Engineering	1	C	1,2	1
9	Case Study: For the Given application identify the stakeholders, gather and classify the requirements according to the types.	1	D,I	1,2	1
	UNIT II: REQUIREMENT INCEPTION AND ELICITATION	**10**			
10	Requirement Inception, Sources of requirements	2	C	1,4	1,5
11	Introduction to requirement Elicitation	1	C	4	1,4
12	Classical Elicitation Techniques – Interview, Questionnaire, Social analysis	2	C,I	4	1,2,4
13	Modern Elicitation Techniques – Brainstorming, Prototyping, Scenarios, User Centered Design, Walkthrough, Use case, Joint Application Development, Requirement Reuse, Feature-Oriented Domain Analysis	4	C,I	4	1,2,4

14	Case Study: For the given application apply various techniques and Elicitate the requirements	1	D,I		1,2
	UNIT III: REQUIREMENT DOCUMENT AND VALIDATION	**10**			
15	Introduction to Requirement Document , Structure of Document	1	C	1,3	2
16	Vision, Scope and Elicitation notes	1	C	1	1,2
17	Requirement Specification techniques	1	C	4	1,2
18	Introduction to requirement validation -Classical Requirement Validation techniques – Inspection, Simple Check, Desk Check, Walkthrough, Formal review - Logical Analysis	3	C,O	3,4	1
19	Prototype & Enactments, Functional test Design	2	C,D	3,4	1
20	Development of User manual	1	C,D	3,4	2
21	Case Study: For the given application validate and document the specifications.	1	D		2
	UNIT IV: REQUIREMENT QUALITY ANALYSIS AND MANAGEMENT	**09**			
22	Business Drivers of Quality - Components of Integrated Quality Approach	1	C	4,5	1
23	Quality Improvement techniques , PDCA Cycle	2	C,I	2,4	1
24	Introduction to Requirement Management- Requirement Identification- Requirements traceability	1	C,D	4	1
25	Requirement Traceability models, Traceability Matrix - Traceability List & Tree	2	D	4	2
26	Introduction to Requirement Traceability- Requirement Traceability methods	1	C	4	2
27	Advanced Traceability, Requirement Change Control	2	C	4	2
	UNIT V: REQUIREMENT ENGINEERING TOOLS	**06**			
28	Doors – Introduction – CASE for Requirement management, DOORS Architecture, History, version Control, Attribute & View	3	C,O	4	2
29	Case Study : Requirement Engineering Tools	3	C,O	4	2
	Total contact hours	**45**			

Note: Students are given individual application, for that they have to do the case study specified in each unit.

LEARNING RESOURCES	
Sl. No.	**TEXT BOOKS**
1	Elizabeth Hull, Ken Jackson and Jeremy Dick, "*RequirementsEngineering",* third edition, Springer, 2010.
2	Ralph R. Young., "*The Requirements EngineeringHandbook"*, 2004.
	REFERENCE BOOKS/OTHER READING MATERIAL
3	Robertson S. & Robertson J, "*Mastering the Requirements Engineering Process: getting requirements right"*, Addison-Wesley, 2013.
4	Sommerville I. &Kotonya G., "*Requirements Engineering: Process and techniques"*, John Wiley, 1998.
5	Pohl, Klaus," Requirements Engineering: Fundamentals, Principles, and Techniques", Springer,2010

Course nature				**Theory**			
Assessment Method (Weightage 100%)							
In-semester	**Assessment tool**	Cycle test I	Cycle test II	Cycle Test III	Surprise Test	Quiz	**Total**
	Weightage	**10%**	**15%**	**15%**	**5%**	**5%**	**50%**
					End semester examination Weightage :		**50%**

15SE205J	Programming In JAVA	L	T	P	C
		3	0	2	4
Co-requisite:	NIL				
Prerequisite:	NIL				
Data Book / Codes/Standards	NIL				
Course Category	P PROFESSIONAL CORE				
Course designed by	Department of Software Engineering				
Approval	32nd Academic Council Meeting, 23rd July, 2016				

PURPOSE	Java is a mature and solidly engineered programming language that is extensively built on object oriented programming concepts. Its in-built security and safety features together support for advanced programming tasks like networking, database connectivity, rich web applications, and mobile applications. This course is designed around the fundamental concepts of Java that enable the students to design and build more complex Java applications.

INSTRUCTIONAL OBJECTIVES		**STUDENT OUTCOMES**						
At the end of the course, student will be able to								
1.	Acquaint themselves with the fundamental concepts and programming environment.	l						
2.	Design classes and efficiently use the IO streams	c						
3.	Implement object oriented concepts like inheritance, reusability, and encapsulation.	i						
4.	Apply custom exceptions and employ concurrency.	b						
5.	Exploit the power of advanced data structures and basic GUI design.	i						

Session	**Description of Topic (Theory)**	**Contact hours**	**C-D-I-O**	**IOs**	**Reference**
	UNIT I: FUNDAMENTALS OF JAVA TECHNOLOGY AND PROGRAMMING	**6**			
1.	Java platform features, Java technologies-JSR, JCP.	1	C	1	1,2
2.	Data types, Key words, Scoping rules	2	C	1	1,2
3.	Automatic Type Conversion , Type Casting and Arrays	1	C	1	1,2
4.	Operators Precedence & Associativity, Expression. Flow control, new featuresfrom Java5 to Java 7	1	C	1	1,2
5.	Enhanced for loop, switch statements, handling Strings, Entry Point for Java Programs	1	C	1	1,2
	UNIT II: CLASSES, OBJECTS AND STREAMS	**12**			
6.	Class fundamentals: Declaring objects, Assigning object reference variable, Methods & Method Signatures, Method retuning Values, Method with parameters, Variable arguments in Java 5	2	D	2	1,2
7.	I/O Basics: Byte stream& Character Stream, Getting user input: Reading console input & Writing console output, Reading and Writing files-new file system API NIO2	2	C	2	3
8.	Constructors: Default Constructor, Parameterized constructor. this keyword, Garbage Collector, finalize() method, Overloading methods and constructors	2	C	2	3
9.	Using object as parameters, returning object in methods, recursion, Access control, static and final keyword	2	C	2	3
10.	Nested and Inner classes , Command Line argument	2	D	2	3
11.	String and String Buffer class, Java Bean standards, Naming conventions	2	D	2	3

	UNIT III: INFORMATION HIDING & REUSABILITY	**9**			
12.	Inheritance basics. Using super, Method Overriding. Constructor call. Dynamic method dispatch	3	D	3	4
13.	Abstract class, Using final with inheritance, Default Package. Path & Class Path EnvironmentVariables	2	D	3	4
14.	Package level access ,Importing Packages, Interface: Multiple Inheritance in Java	2	D	3	4
15.	Extending interface, Wrapper Class, Auto Boxing	2	D	3	4
	UNIT IV: EXCEPTION , CONCURRENCY, ENUMERATION AND ANNOTATIONS	**9**			
16.	Exception handling mechanism. new look try/catch mechanism in Java 7	2	C	4	3
17.	Thread class & Runnable Interface. Inter Thread Communication, Synchronization of threads using Synchronized keyword and lock method	2	I	4	3
18.	Thread pool and Executors framework, Futures and callable, Fork-Join in Java. Deadlock conditions	2	I	4	3
19.	Enumeration in Java 5 - usage. Annotations: basics of annotation.	1	C	4	3
20.	The Annotated element Interface. Using Default Values, Marker Annotations. Single-Member Annotations. The Built-In Annotations-Some Restrictions.	2	I	4	3
	UNIT V: GENERICS , COLLECTIONS FRAME WORK AND GUI PROGRAMMING	**9**			
21.	Generics: Basics , Generics and type safety	2	I	5	4
22.	Collections Interfaces – Collection, Set, List, Queue, Collections Classes – Array List, Hash Set, Tree Set. Accessing a Collection via Iterators. Map Interfaces. Map Classes – Abstract Map, Hash Map, Tree Map	2	I	5	4
23.	Introduction to Swing, MVC Connection, Containers – Jframe, JDialog, JPanel,JRootPane, JLayeredPane	2	I	5	4
24.	Placing components into containers, Event Handling, Components – Jbutton, JLabel, JTextField, JComboBox, JList, JTable, JTabbedPane	3	I	5	4
	Total contact hours	**45**			

Sl. No.	Description of experiments	Contact hours	C-D-I-O	IOs	Reference
1.	Program to implement Operators,Flow Controls concepts	3	I	1	1-5
2.	Program to implement Classes, Constructors, Overloading and Access Control	3	I	2	1-5
3.	Program using Nested & Inner Classes, Static and Final	3	I	2	1-5
4.	Program using File Streams and IO Streams	3	I	2	1-5
5.	Program to implement Strings, String Buffer Concept	3	I	2	1-5
6.	Program using Interfaces, Abstract Classes	3	I	3	1-5
7.	Program to implements Exceptions Concepts	3	I	4	1-5
8.	Program using Threads	3	I	4	1-5
9.	Program using Collections, Generics concepts	3	I	5	1-5
10.	Program to implement Swing Application	3	I	5	1-5
	Total contact hours	30			

LEARNING RESOURCES	
Sl. No.	**TEXT BOOKS**
1.	Herbert Schildt, "The Complete Reference (Fully updated for jdk7)", Oracle press Ninth Edition,2014.
	REFERENCE BOOKS/OTHER READING MATERIAL
2.	Cay S. Horstmann, "Core Java Volume –I Fundamentals", Prentice Hall , 10th Edition, 2015.
3.	Deitel&Deitel, "Java How to Program", Prentice Hall, 10th Edition, 2016.
4.	Herbert Schildt ,"Java: A Beginner's Guide", Sixth Edition, Oracle Press, 2014.
5.	https://docs.oracle.com/javase/tutorial

Course nature				**Theory + Practical**			
Assessment Method – Theory Component (Weightage 50%)							
In-semester	**Assessment tool**	Cycle test I	Cycle test II	Cycle Test III	Surprise Test	Quiz	**Total**
	Weightage	**10%**	**15%**	**15%**	**5%**	**5%**	**50%**
						End semester examination Weightage :	**50%**
Assessment Method – Practical Component (Weightage 50%)							
In-semester	**Assessment tool**	Experiments	Record	MCQ/Quiz/Viva Voce		Model examination	**Total**
	Weightage	**40%**	**5%**	**5%**		**10%**	**60%**
						End semester examination Weightage :	**40%**

15SE311	Software Architecture And Design	L	T	P	C
		3	0	0	3
Co-requisite:	NIL				
Prerequisite:	15SE202				
Data Book / Codes/Standards	NIL				
Course Category	P PROFESSIONAL CORE				
Course designed by	Department of Software Engineering				
Approval	32nd Academic Council Meeting, 23rd July, 2016				

PURPOSE	To provide knowledge and expose students to the activities in software design covering design conceptualizations, strategies, architectures, human interfaces, design methods and tools

INSTRUCTIONAL OBJECTIVES		**STUDENT OUTCOMES**						
At the end of the course, student will be able to								
1.	Describe different approaches to design software application	c						
2.	Analyze specifications and identify appropriate design strategies.	c						
3.	Develop an appropriate design for a given set of requirements	c						
4.	Identify applicable design patterns for the solution	e						
5.	Abstract and document reusable design patterns	k						
6.	Evaluate a given design against the specifications	h	k					

Session	Description of Topic	Contact hours	C-D-I-O	IOs	Reference
	Unit 1:Software Design Concepts	**9**			
1.	Introduction - Definitions of design	1	C	1-4	1-3
2.	Fundamental design issues (e.g. persistent data, storage management, exceptions, etc).	2	C	1	1
3.	Context of design within multiple software development life cycles. Design principles (information hiding, cohesion and coupling). Interactions between design and requirements	2	C,D	1	1,2
4.	Design for quality attributes (e.g. reliability, usability, maintainability, performance, testability, security, fault tolerance, etc).	2	C,D	1	2
5.	Design tradeoffs	2	C	1	1,2,3
	UNIT 2 Software Design strategies	**9**			
6.	Role of design strategy - Describing the design process the D-Matrix	2	C	4,5	2
7.	Design by top-down decomposition -Design by composition	3	C	4,5	2
8.	Function-oriented design- Object-oriented design	2	D	4,5	1,2
9.	Data-Centered design-Aspect oriented design	2	C	4,5	1, 2
	UNIT 3: Software Architectural design	**9**			
10.	Software Architectural Patterns & Styles	2	C	3	1,5
11.	Requirements traceability in architecture	2	C,D	3	1
12.	Service-Oriented Architectures	2	C,D	3	1
13.	Architectures for network, mobile, and embedded systems	2	C,I	2,3	9
14.	Design principles of human computer interaction	1	C,I	2,3	1
	Unit 4: Software detailed design& patterns	**9**			
15.	Detailed design	2	C	3	1
16.	Design patterns	2	C,D,I	5	1,2,4,10
17.	Database design	2	D	4	7,9
18.	Design of networked and mobile systems	2	D	4	10
19.	Design notations (e.g., class and object diagrams, UML, state diagrams, and formal specification)	1	I	5	1,3,6,8

	UNIT 5: Software Design Evaluation and Metrics	**9**			
20.	Design attributes (e.g., coupling, cohesion, information hiding, and separation of concerns)	3	D,I	6	2
21.	Design metrics	3	C	6	3
22.	Formal design analysis	3	C	6	2
	Total contact hours	**45**			

LEARNING RESOURCES	
Sl. No.	**TEXT BOOKS**
1.	Carlos Otero, "Software Engineering Design: Theory and Practice", CRC Press, 2012
2.	David Budgen, *Software Design (2nd Edition),* Addison Wesley, 2012
3.	Hassan Gomma, "Software Modeling and design with UML", Cambridge University Press,2010
	REFERENCE BOOKS/OTHER READING MATERIAL
4.	Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1995 (Classic and famous seminal book for software design patterns.)
5.	Eric J. Braude, *Software Design: From Programming to Architecture (First Edition),* Wiley, 2003
6.	Bernd Bruegge, Allen H. Dutoit, *Object Oriented Software Engineering Using UML, Patterns and Java (Second Edition),* Prentice Hall, 2003
7.	"Database SystemConcepts"Henry F. Korth, Abraham Silberschatz, Sudarshan S,6th Edition, Tata McGraw-HillEducation, 2010
8.	R. S. Pressman, "Software Engineering", Fifth Edition, McGraw Hill Inc., 2001
9.	"" Data base Systems: A practical approach to design implementation and management" Thomas Connolly and Carolyn, Begg, AddisionWesleyFourth edition, 2010
10.	"Elemental design Patterns", Jason McC. Smith, Pearson,2012

Course nature				**Theory**			
Assessment Method (Weightage 100%)							
In-semester	**Assessment tool**	Cycle test I	Cycle test II	Cycle Test III	Surprise Test	Quiz	**Total**
	Weightage	**10%**	**15%**	**15%**	**5%**	**5%**	**50%**
					End semester examination Weightage :		**50%**

15SE390L	Industrial Training I	L	T	P	C
		0	0	3	2
Co-requisite:	NIL				
Prerequisite:	NIL				
Data Book / Codes/Standards	NIL				
Course Category	P PROFESSIONAL CORE				
Course designed by	Department of Software Engineering				
Approval	32nd Academic Council Meeting ,23rd July 2016				

PURPOSE	To provide short-term work experience in an Industry/ Company/ Organisation							
INSTRUCTIONAL OBJECTIVES		**STUDENT OUTCOMES**						
At the end of the course, student will be able								
1.	To get an inside view of an industry and organization/company				j			
2.	To gain valuable skills and knowledge				j			
3.	To make professional connections and enhance networking	f	g					
4.	To get experience in a field to allow the student to make a career transition			i				

Session	Description of Topic	Contact hours	C-D-I-O	IOs	Reference
	1. It is mandatory for every student to undergo this course. 2. Every student is expected to spend a minimum of 15-days in an Industry/ Company/ Organization, during the summer vacation. 3. The type of industry must be NOT below the Medium Scale category in his / her domain of the degree programme. 4. The student must submit the "Training Completion Certificate" issued by the industry / company / Organisation as well as a technical report not exceeding 15 pages, within the stipulated time to be eligible for making a presentation before the committee constituted by the department. 5. The committee will then assess the student based on the report submitted and the presentation made. 6. Marks will be awarded out of maximum 100. 7. Appropriate grades will be assigned as per the regulations. 8. Only if a student gets a minimum of pass grade, appropriate credit will be transferred towards the degree requirements, as per the regulations. 9. It is solely the responsibility of the individual student to fulfill the above conditions to earn the credits. 10. The attendance for this course, for the purpose of awarding attendance grade, will be considered 100%, if the credits are transferred, after satisfying the above (1) to (8) norms; else if the credits are not transferred or transferable, the attendance will be considered as ZERO. 11. The committee must recommend redoing the course, if it collectively concludes, based on the assessment made from the report and presentations submitted by the student, that either the level of training received or the skill and / or knowledge gained is NOT satisfactory.		D, I,O	1,2,3,4	

Course nature		**Training – 100% internal continuous assessment**		
Assessment Method (Weightage 100%)				
In-semester	**Assessment tool**	**Presentation**	**Report**	**Total**
	Weightage	**80%**	**20%**	**100%**
End semester examination Weightage :				**0%**

15SE312	Software Modeling And Analysis	L	T	P	C
		3	0	0	3
Co-requisite:	NIL				
Prerequisite:	15SE202				
Data Book / Codes/Standards	NIL				
Course Category	P PROFESSIONAL CORE				
Course designed by	Department of Software Engineering				
Approval	32nd Academic Council Meeting, 23rd July, 2016				

PURPOSE	To model the software system and analyze its characteristics and correctness.							
INSTRUCTIONAL OBJECTIVES		**STUDENT OUTCOMES**						
At the end of the course, student will be able to								
1	Select a suitable modeling method according to problem area and assignment, and can justify their choice.	b	a					
2	Formulate models of a system to describe the system on different levels of abstraction and from different viewpoints.	c	a	k				
3	Apply the Formal models in the software development	a						
4	Apply the model checking concepts in tool	e		k				

Session	Description of Topic	Contact hours	C-D-I-O	IOs	Reference
	UNIT I: Engineering System Requirements	8			
1.	Introduction to software modelling	1	C	1	1
2.	Modeling principles	1	C	1	1
3.	Goal features as model annotations,Goal refinement	1	C	1	2
4.	Representing conflicts among goals,Connecting the goal model with other system views.	1	C	1	2
5.	Modelling alternative options, Goal diagrams as AND/OR graphs.	1	C	1	2
6.	Documenting goal refinements and assignments with annotations	1	C	1	2
7.	Building goal models: Heuristic rules and reusable patterns	1	C,D	1	2
8.	Analyzing System Risks: Modelling obstacles, Obstacle analysis for a more robust goal model.	1	C,D	1	2
	UNIT II Information Modeling	9			
9.	Representing domain concepts by conceptual objects	2	C	2	1
10.	Entities, Associations, Attributes.	2	C	2	1
11.	Built–in associations for structuring object models	1	C,D	2	1
12.	Class Diagrams, Heuristic rules for building object models	1	C	2	1
13.	ER Diagrams	1	C,D	2	1
14.	Case Study : ER Diagrams, Class Diagrams	2	C,D	2	1
	UNIT III Behavioral Modeling	8			
15.	Characterizing system agents	1	C	2	1
16.	Building agent models	1	C,D	2	1
17.	Goals, agents, objects and operations : Overview	1	C	2	1
18.	Modelling instance behaviors	2	C	2	1
19.	Modelling class behaviors	1	C	2	1
20.	Building behavior models	2	C,D	2	1

	UNIT IV Formal Modeling and Analysis	10			
21.	Correctness of Object Oriented Programs Design by Contract, The Class Invariant	1	C	1	1,2
22.	Example: Correctness of a Stack Application	1	C,D	1	2
23.	Formal Specification of System Models	1	C	3	1
24.	A real–time temporal logic for specifying model annotations	1	C,I	3	1
25.	Specifying goals in the goal model.	1	C,I	3	1
26.	Specifying descriptive properties in the object model	1	C	3	1
27.	Specifying operationalization's in the operation model	1	C	3	1
28.	Case Study : Formal Reasoning for Specification Construction and Analysis	1	C	3	1
29.	Formal conflict analysis.	1	C,O	3	1
30.	Synthesizing behaviour models for animation and model checking.	1	C	3	1
	UNIT V Static Analysis	10			
31.	Introduction to Kripke Structures,	1	C	4	3
32.	Modeling System Design as Kripke Structure	1	C,D	4	3
33.	Exercise on Kripke Modelling, Recap to Kripke Structures	1	C	4	3
34.	Exercises on system modeling using kripke structures, Introduction to LTL	1	D,I	4	3
35.	Properties of a system, Liveness and Safety properties , Exercise on specifying properties as LTL formula	1	C	4	3
36.	Introduction to NuSMV tool, Model Checking using NuSMV,Introduction to SPIN CHECKER tool	1	C	4	3
37.	Model Checking using SPIN checker tool	2	C,I	4	3
38.	System Property as first order logic formula Proof of correctness using theorem prover	1	C	4	3
39.	Introduction Isabelle tool, Theorem proving using Isabelle	1	C	4	3
	Total contact hours	**45**			

LEARNING RESOURCES	
Sl. No.	**TEXT BOOKS**
1.	Axel van Lamsweerde "Requirements Engineering: From System Goals to UML Models to Software Specifications" ISBN : 978-0-470-01270-3 February 9, 2009 Wiley
2.	http://www.bowdoin.edu/~allen/courses/cs260/readings/ch12.pdf
3.	Gerard J. Holzmann, "The SPIN Model Checker: Primer and Reference Manual" ISBN-13: 978-0321773715 , AT&T Bell Labs Murray Hill New Jersey ©2004 | Addison-Wesley Professional

Course nature				**Theory**			
Assessment Method (Weightage 100%)							
In-semester	**Assessment tool**	Cycle test I	Cycle test II	Cycle Test III	Surprise Test	Quiz	**Total**
	Weightage	**10%**	**15%**	**15%**	**5%**	**5%**	**50%**
End semester examination Weightage :							**50%**

15SE313	Software Project Management	L	T	P	C
		3	0	0	3
Co-requisite:	NIL				
Prerequisite:	15SE202				
Data Book / Codes/Standards	NIL				
Course Category	P PROFESSIONAL CORE				
Course designed by	Department of Software Engineering				
Approval	32nd Academic Council Meeting, 23rd July, 2016				

PURPOSE	This course on Software Project Management highlights Software Project planning approaches, methodologies and standards of software development and management							
INSTRUCTIONAL OBJECTIVES		**STUDENT OUTCOMES**						
At the end of the course, student will be able to								
1.	To provide students with a clear understanding of the unique risks, issues, and critical success factors associated with technology projects	d						
2.	To introduce students to the role and function of project management	d						
3.	To familiarize the process of project life cycle	e						
4.	To understand the various techniques for requirements, planning and managing a technology project	a	k					
5.	To examine basic methodologies for software design, development, testing, closure and implementation	a	d	e				
6.	To examine various techniques for managing a software development team	g	k					
7.	To understand the need and techniques for managing users and user expectations	g	h	j				
8.	To acquire the latest industry knowledge, tools and comply to the latest global standards for project management	i	k					

Session	Description of Topic	Contact hours	C-D-I-O	IOs	Reference
	UNIT I: PROJECT MANAGEMENT FUNDAMENTALS	**12**			
1.	Introduction to Software Project Management – all life cycle activities	1	C	2,5	1-7
2.	Project Initiation Management – scope, objective,size and factors	1	C	2,4	1,2,3,4,6
3.	Software Project Effort and Cost Estimation – cocomo, artifacts	2	C,D	1	1,2,3,4,6
4.	Risk Management : Perform The risk analysis for the given case study	2	C,D	1	1,2,3,4,6
5.	Configuration Management – techniques	2	C	3	1,2
6.	Project Planning – wbc, planning,scope and risk	2	C	4	1,2
7.	Project Monitoring and Control – measuring task, status report, evm	1	C	6	1,2
8.	Project Closure – closure steps	1	C	6,7	1,2
	UNIT II: SOFTWARE LIFE-CYCLE MANAGEMENT	**10**			
9.	Introduction to Software Life-Cycle Management – life cycle process	2	C C	3	1,2,3,4,6
10.	Software Requirement Management – requirement and management	2	C,D	4	1,2,3
11.	Software Design Management – standards, techniques	2	C, D	5	1,2,3

12.	Software Construction – reviews, walkthrough, inspections	1	C	5	1,2
13.	Software Testing – Verification, validation, strategy, automation and monitoring	1	C	5	1,2
14.	Product Release and Maintenance – types and techniques	2	C	5	1,2
	UNIT III: SOFTWARE ENGINEERING MANAGEMENT	**7**			
15.	Process Standards Introduction root causes, solution of s/w problem,	2	C	7	1,2,5
16.	Software Process Standards and Process Improvement – CMM, ISO, IEEE a nut shell	3	C	7	1, 2,5,1
17.	Process Selection – Agile, Traditional method to Agile	2	C	3, 7	1, 2,5,7
	UNIT IV: PEOPLE MANAGEMENT	**8**			
18.	Introduction to People Management – people, team and supplier management	2	C	6, 7	1,2
19.	Team Management – organizational structure, team effectiveness	2	C	6, 7	1,2
20.	Customer Management – expectation and negotiation	2	C	6, 7	1,2
21.	Supplier Management – agreement and communication	2	C	6, 7	2,4,5
	UNIT V: TOOLS AND TECHNIQUES	**8**			
22.	Software Project Management Tools Introduction – tools application, cost and effectiveness	2	C	8	2,3,4
23.	Project Management and Software Life-Cycle Tools – life cycle and project management templates	2	C	8	1,2,4
24.	Software Project Templates – WBS and monitoring tools	2	C	8	2,3
25.	Future Tools and Techniques – open source, asp, service provider, trends in s/w industry	2	C	8	1,2
	Total contact hours	45			

LEARNING RESOURCES	
Sl. No.	**TEXT BOOKS**
1.	Ashfaque Ahmed, "Software Project Management: a process-driven approach",Boca Raton, Fla. : CRC Press, 2012.
2.	Bob Hughes, Mike Cottere, Rajib Mall, "Software Project Management", 5th Ed, Tata McGraw Hill, 2011.
	REFERENCE BOOKS/OTHER READING MATERIAL
3.	Ramesh, Gopalaswamy, "Managing Global Projects", Tata McGraw Hill, 2005.
4.	Royce, "Software Project Management", Pearson Education, 1999.
5.	Jalote, "Software Project Management in Practice", Pearson Education, 2002.
6.	Rapid Development, McConnell, Steve, Microsoft Press, 1996, ISBN 1-55615-900-5.
7.	Information Technology Project Management, Schwalbe, Kathy, 2nd ed., Course Technology, 2002, ISBN 0-619-03528-5 (this text includes a 120 day evaluation copy of MS-Project 2000).

Course nature				**Theory**			
Assessment Method (Weightage 100%)							
In-semester	**Assessment tool**	Cycle test I	Cycle test II	Cycle Test III	Surprise Test	Quiz	**Total**
	Weightage	**10%**	**15%**	**15%**	**5%**	**5%**	**50%**
					End semester examination Weightage :		**50%**

15SE314J	**Software Verification And Validation**	**L**	**T**	**P**	**C**

		3	**0**	**2**	**4**
Co-requisite:	NIL				
Prerequisite:	15SE202				
Data Book / Codes/Standards	NIL				
Course Category	P PROFESSIONAL CORE				
Course designed by	Department of Software Engineering				
Approval	32nd Academic Council Meeting, 23rd July, 2016				

PURPOSE	The purpose of this course is to provide an understanding of concepts and techniques for testing software and assuring software component or system satisfies its requirements and meets stakeholder expectations.

INSTRUCTIONAL OBJECTIVES		**STUDENT OUTCOMES**						
At the end of the course, student will be able to								
1	Design test cases suitable for a software development in various domains	a						
2	Identify suitable tests to be carried out	b	e					
3	Prepare test planning based on the document ,design test plans and document test plans	b	d					
4	Develop and validate a test plan	b	e					
5	Use automation testing tools	b	k					

Session	Description of Topic (Theory)	Contact hours	C-D-I-O	IOs	Reference
	UNIT 1 - Fundamentals of Testing	**9**			
1.	Necessity of Testing-Case Studies on "Impact of Software bugs", Objectives of Testing, Basics Definitions	2	C	1	1,3,4
2.	Testing Principles-Illustrations, Fundamental Test Process, The tester's role in a software development organization, Test planning	2	C	1,2	1,2,3,4
3.	Establishing Test Policy, Structured approach to testing Test Factors and **Eleven Step software testing process**	1	C	1,2	3,4
4.	Origin of Defects, Defect Repository and Test Design, Defect Examples, Case Studies – "Identify the defect"	2	C	2	3,4
5.	Developer/Tester support of developing a defect repository, Defect Analysis and Prevention Strategies	2	C	2	3,4
	UNIT 2 - Analysis and Test Case Design	**7**			
6.	Test Case Design Strategies Introduction to basic design strategies	1	C	1	1,2,3
7.	White Box Strategies - Peer Reviews, Inspections, Walkthrough, Static Analysis Tools, Code functional Testing, Coverage and Control Flow Graphs, Paths Code Complexity, Evaluating test adequacy criteria, Case Studies-"Applying the suitable White Box Strategy"	3	C,D,I	1,2,3	1,2,3
8.	Black Box Testing Strategies: Requirements Based Testing, Boundary Value Analysis, Equivalence Class Partitioning, Cause Effect graphing, Random Testing, Evaluating test adequacy criteria. Case Studies-"Applying suitable Black Box Testing Strategy"	3	C,D,I	1,2,3	1,2,3
	UNIT 3 - Levels of Testing	**10**			
9.	The need for levels of testing, Unit Testing: Planning, Test Harness, Running the tests Recording Results	1	C,D	1,2,3	1,2
10.	Integration Testing: Goals, Design and Plan	1	C,D	1,2	1,2
11.	System Testing goals, Types of System Testing:	2	C,D	1,2	1,2

	Functional, Performance, Stress, Configuration, Stress, Security, Recovery, Reliability and Usability Testing				
12.	Regression Testing Alpha, Beta and Acceptance Testing	2	C,D	1,2	1,2
13.	Role of use cases in testing, Applying Testing Skills: Compatibility testing, Internationalization testing	2	C,D	1,2	1,2
14.	Applying Testing Skills, Testing Documentation plan, Recording test cases, Reporting and Measurement of Success	2	C,D,I	1,2,3	1,2
	UNIT 4: Test Management and Defect Analysis	**12**			
15.	Test Management: Choice of Standards, Infrastructure Management, Test People Management	2	C	1	1,2
16.	Test Plan Components, Test Plan Attachments	2	C,D	3,4	1,2
17.	Locating Test Items, Managing Issues, Addressing Perception, Taking team together	2	C,D	2,3	1,2
18.	Focus on technology, process and management, Customer perspective, Documentation uses and Documentation Types	3	C,D	3,4	1,2
19.	Test Analysis report Documentation, Analyze reports, problem tracking, Controlling and Monitoring Test Progress	3	C,D,I	3,4	1,2
	UNIT 5 - Test Automation	**7**			
20.	Software Test Automation: Skills and Scope Design and Architecture for Automation, Challenges in Automation	1	C	3,5	1,2,5
21.	Requirements for a test tool, Process Model for Automation, Selecting the test tool	1	C,I	5	1,2,5
22.	Test Metrics and measurements: Role, need and types	2	C	5	1,2
23.	Project Metrics, Progress Metrics, Productivity Metrics	2	D,I,O	1,2,3	1,2
24.	The Future: Software Quality Assurance Career as a Software Tester	1	D,I,O	1,2,3,4,5	1,2
	Total contact hours	**45**			

Sl. No.	Description of experiments	Contact hours	C-D-I-O	IOs	Reference
1.	For the given application design adhoc test cases to test the system	2	D,I	1,2,3	1,2
2.	Design the test cases to test the given system using following Black Box testing technique: a. Boundary Value Analysis, Worst Boundary Value Analysis, Robust Boundary Value Analysis, Robust Worst Boundary Value Analysis b. Equivalence class testing (Input/Output domain) c. Decision table and cause-effect graph	2	D,I	1,2,3	1,2
3.	For a given program segment using tool a. Draw the control flow graph b. Determine the cyclomatic complexity c. Determine the independent paths d. Generate the test cases for each independent path How many tests are required for 100% decision coverage? Give the test cases.	2	D,I	1,2,3	1,2
4.	Perform functional testing for the given application/project using the testing tool	6	D,I,O	5	1,2,3
5.	Perform web testing for the given application/project using the testing tool	4	D,I,O	5	1,2,3

6.	Perform load testing for the given application/project using the testing tool	4	D,I,O	5	1,2,3
7.	Perform bug tracking for the given application/project using tool	5	D,I,O	5	1,2,3
8.	Perform test management for the given application/project using testing tool	5	D,I,O	3,4,5	1,2,3
	Total contact hours	**30**			

LEARNING RESOURCES

Sl. No.	**TEXT BOOKS**
1.	Srinivasan Desikan and Gopalaswamy Ramesh, "Software Testing – Principles and Practices", Pearson Education, 2006.
2.	Ron Patton, "Software Testing", Second Edition, Sams Publishing, Pearson Education, 2007.
	REFERENCE BOOKS/OTHER READING MATERIAL
3.	Ilene Burnstein, "Practical Software Testing", Springer International Edition, 2003.
4.	Aditya P. Mathur, "Foundations of Software Testing _ Fundamental Algorithms and Techniques", Dorling Kindersley (India) Pvt. Ltd., Pearson Education, 2008.
5.	RenuRajani, Pradeep Oak, "Software Testing-Effective Methods, Tools and Techniques", Tata McGraw Hill Education, 2011.

Course nature				**Theory + Practical**			
Assessment Method – Theory Component (Weightage 50%)							
In-semester	**Assessment tool**	Cycle test I	Cycle test II	Cycle Test III	Surprise Test	Quiz	**Total**
	Weightage	**10%**	**15%**	**15%**	**5%**	**5%**	**50%**
					End semester examination Weightage :		**50%**
Assessment Method – Practical Component (Weightage 50%)							
In-semester	**Assessment tool**	Experiments	Record	MCQ/Quiz/Viva Voce		Model examination	**Total**
	Weightage	**40%**	**5%**	**5%**		**10%**	**60%**
					End semester examination Weightage :		**40%**

15SE375L	Minor Project I	L	T	P	C
		0	0	3	2
Co-requisite:	NIL				
Prerequisite:	NIL				
Data Book / Codes/Standards	NIL				
Course Category	P PROFESSIONAL				
Course designed by	Department of Software Engineering				
Approval	32nd Academic Council Meeting ,23rd July 2016				

PURPOSE	To obtain an hands-on experience in converting a small novel idea / technique into a working model / prototype involving multi-disciplinary skills and / or knowledge and working in at team.

INSTRUCTIONAL OBJECTIVES		**STUDENT OUTCOMES**						
At the end of the course, student will be able								
1.	To conceptualise a novel idea / technique into a product	c						
2.	To think in terms of multi-disciplinary environment		d					
3.	To understand the management techniques of implementing a project				k			
4.	To take on the challenges of teamwork, prepare a presentation in a professional manner, and document all aspects of design work.			g				

Session	Description of Topic	Contact hours	C-D-I-O	IOs	Reference
	An Multidisciplinary project to be taken up by a team of maximum of ten students. Development of prototype product, a 3D model, simulation, blueprint for a larger project and any other development work are permitted. The contribution of the individuals in the project should be clearly brought out. A combined report is to be submitted. A presentation is to be made for the reviewers on the work done by the candidate.		C,D,I	1,2,3,4	

Course nature		**Project – 100% internal continuous assessment**	
Assessment Method (Weightage 100%)			
In-semester	**Assessment tool**	Refer the table	**Total**
	Weightage	Refer the table below	**100%**
End semester examination Weightage :			**0%**

Assessment components

Assessment component	Expected outcome	Evaluators	Criteria or basis	Marks
Project proposal (Review – I)	A short presentation to be delivered on: • A brief, descriptive project title (2-4 words). This is critical! • The 3 nearest competitors (existing solutions) and price. • Team members name, phone number, email, department/degree program, and year. • A description of the product opportunity that has been identified. To include: Documentation of the market need, shortcomings of existing competitive products, and definition of the target market and its size. • Proposed supervisor / guide	Panel of reviewers	Viability / feasibility of the project Extent of preliminary work done.	**0**

Review II	• Mission Statement / Techniques • Concept Sketches, Design Specifications / Modules & Techniques along with System architecture • Coding	Panel of reviewers	Originality, Multi-disciplinary component, clarity of idea and presentation, team work, handling Q&A.	**20**
Review III	• Final Concept and Model / Algorithm/ Technique • Drawings, Plans / programme output • Financial Model / costing • Prototype / Coding • Final Presentation and Demonstration	Panel of reviewers	Originality, Multi-disciplinary component, clarity of idea and presentation, team work, handling Q&A.	**50**
Final technical Report	A good technical report	Supervisor / Guide	Regularity, systematic progress, extent of work and quality of work	**30**
			Total	**100**

15SE380L	Seminar I	L	T	P	C
		0	0	3	2
Co-requisite:	NIL				
Prerequisite:	NIL				
Data Book / Codes/Standards	NIL				
Course Category	P PROFESSIONAL				
Course designed by	Department of Software Engineering				
Approval	32nd Academic Council Meeting , 23rd July 2016				

PURPOSE	To inculcate the research culture among the students through literature reading, modelling a problem, analyzing and presenting.							
INSTRUCTIONAL OBJECTIVES		**STUDENT OUTCOMES**						
At the end of the course, student will be able								
1.	To understand the research methodology adopted by various researchers	h	i	j				
2.	To mathematically model a problem, critically analyse it and adopt strategies to solve	b	c	e				
3.	To understand and present a well documented research	e	g					

Session	Description of Topic	Contact hours	C-D -I-O	IOs	Reference
	Guidelines for conducting 15SE80L Seminar for B.Tech 1. Upon registering for the course the student must identify a sub-domain of the degree specialization that is of interest to the student and start collecting research papers as many as possible. 2. After collecting sufficient number of research papers the student must peruse all the papers, meet the course faculty and discuss on the salient aspects of each and every paper. 3. The course faculty, after discussion with the student will approve TWO research papers that is appropriate for presentation. 4. The student must collect additional relevant reference materials to supplement and compliment the two research papers and start preparing the presentation. 5. Each student must present a 15-minute presentation on each of the approved research paper to the panel of evaluators. 6. The presenter must present one research paper within the first half of the semester (6 weeks) and another research paper in the next half of the semester (6 weeks) as per the schedule. 7. All other students registered for the course will form the audience. 8. The audience as well as the evaluators will probe the student with appropriate questions and solicit response from the presenter. 9. The presentation will be evaluated against 7 to 8 assessment criteria by 4 to 5 evaluators. 10. The score obtained through the presentations of TWO research papers will be converted to appropriate percentage of marks. This course is 100% internal continuous assessment.		C,D	1,2,3,4	
	Total contact hours	**30**			

Course nature		**100% internal continuous assessment.**		
Assessment Method (Weightage 100%)				
In-Semester	**Assessment tool**	Presentation 1	Presentation 2	**Total**
	Weightage	**50%**	**50%**	**100%**
End semester examination Weightage :				**0%**

EVALUATION OF SEMINAR PRESENTATIONS

Name of the Student: Date:

Register Number: Degree and Branch:

Topic:

Sl. No.	Criteria for Assessment	Evaluator 1	Evaluator 2	Evaluator 3	Evaluator 4	Evaluator 5
1	Understanding of the subject					
2	Clarity of presentation					
3	Appropriate use of Audio visual aids					
4	Whether cross references have been consulted					
5	Ability to respond to questions on the subject					
6	Time scheduling					
7	Completeness of preparation					

Poor	1	Below Average	2	Average	3	Good	4	Very Good	5

Overall Grades:

Remarks:

Signature of Course Coordinator

15SE385L	MOOCs I	L	T	P	C
		0	0	3	2
Co-requisite:	NIL				
Prerequisite:	NIL				
Data Book / Codes/Standards	NIL				
Course Category	P PROFESSIONAL				
Course designed by	Department of Software Engineering				
Approval	32nd Academic Council Meeting, 23rd July 2016				

PURPOSE	To offer students the opportunity to study with the world's best universities by integrating select MOOCs in a regular degree programme and providing students full credit transfer, as per university regulations, if they earn a "Verified / Completion Certificate" and take a proctored examination through a secure, physical testing center.

INSTRUCTIONAL OBJECTIVES		**STUDENT OUTCOMES**						
At the end of the course, student will be able								
1.	To apply the concepts, theories, laws, technologies learnt herein to provide engineering solutions.	f	h	i	j			

Course nature		**Online - 100% internal continuous assessment.**				
Assessment Method (Weightage 100%)						
In-semester	**Assessment tool**	Quiz	Assignment	Non-proctored / Unsupervised Tests	Proctored / Supervised Test	**Total**
	Weightage	**25%**	**25%**	**10%**	**40%**	**100%**
End semester examination Weightage :						**0%**

Registration process, Assessment and Credit Transfer:

1. Students can register for courses offered by approved global MOOCs platforms like edX, Coursera or Universities with which SRM partners specifically for MOOCs.
2. Annually, each department must officially announce, to the students as well as to the Controller of Examinations, the list of courses that will be recognized and accepted for credit transfer.
3. The department must also officially announce / appoint one or more faculty coordinator(s) for advising the students attached to them, monitoring their progress and assist the department in proctoring the tests, uploading the marks / grades, and collecting and submitting the graded certificate(s) to the CoE, within the stipulated timeframe.
4. Student who desires to pursue a course, from the above department-approved list, through MOOCs must register for that course during the course registration process of the Faculty of Engineering and Technology, SRM University.
5. The maximum credit limits for course registration at SRM will include the MOOCs course registered.
6. The student must periodically submit the marks / grades obtained in various quizzes, assignments, tests etc immediately to the Faculty Advisor or the Course Coordinator for uploading in the university's academic module.
7. The student must take the final test as a Proctored / Supervised test in the university campus.
8. The student must submit the "Certificate of Completion" as well as the final overall Marks and / or Grade within the stipulated time for effecting the grade conversion and credit transfer, as per the regulations. It is solely the responsibility of the individual student to fulfil the above conditions to earn the credits.
9. The attendance for this course, for the purpose of awarding attendance grade, will be considered 100% , if the credits are transferred, after satisfying the above (1) to (7) norms; else if the credits are not transferred or transferable, the attendance will be considered as ZERO.

15SE490L	Industrial Module I	L	T	P	C
		0	0	3	2
Co-requisite:	NIL				
Prerequisite:	NIL				
Data Book / Codes/Standards	NIL				
Course Category	P PROFESSIONAL				
Course designed by	Department of Software Engineering				
Approval	32nd Academic Council Meeting , 23rd July 2016				

PURPOSE	To offer students the opportunity to interact with industries and learn the best practices adopted by them.

INSTRUCTIONAL OBJECTIVES		**STUDENT OUTCOMES**						
At the end of the course, student will be able								
1.	To obtain an insight into the current industrial trends and practices	h						
2.	To obtain an insight into the technologies adopted by industries	k						
3.	To obtain an insight into the technical problems encountered by the industries and the scope for providing solutions.	e						
4.	To network with industry	h						

Description of Topic	**Contact hours**	**C-D-I-O**	**IOs**	**Reference**
1. The department will identify and shortlist few emerging topics that are trending in industry. 2. The department will identify experts from industry who are willing to deliver modules on the shortlisted topics. 3. The identified expert will assist the department in formulating the course content to be delivered as a 30-hour module, prepare lectures notes, ppt, handouts and other learning materials. 4. The department will arrange to get the necessary approvals for offering the course, from the university's statutory academic bodies well before the actual offering. 5. The department must officially announce, to the students as well as to the Controller of Examinations, the list of courses that will be offered as industry module. 6. The department must also officially announce / appoint one or more faculty coordinator(s) for advising the students attached to them, monitoring their progress and assist the department in proctoring/supervising/assessment the quizzes, assignments, tests etc, uploading the marks, attendance etc, within the stipulated timeframe. 7. The Student who desires to pursue a course, from the above department-approved list, must register for that course during the course registration process of the Faculty of Engineering and Technology, SRM University. 8. The maximum credit limits for course registration at SRM will include the Industry Module also. 9. All academic requirements of a professional course like minimum attendance, assessment methods, discipline etc will be applicable for this Industry Module. 10. The course will be conducted on weekends or beyond the college regular working hours.		C,D,I,O	1,2,3,4	
Total contact hours	**30**			

Course nature	**100% internal continuous assessment.**

Assessment Method – Theory Component (Weightage 50%)							
In-semester	Assessment tool	Cycle test I	Cycle test II	Cycle Test III	Surprise Test	Quiz	Total
	Weightage	10%	15%	15%	5%	5%	50%
End semester examination Weightage							50%

15SE411	Security In Networks And Software Development	L	T	P	C
		3	0	0	3
3Co-requisite:	NIL				
Prerequisite:	NIL				
Data Book / Codes/Standards	NIL				
Course Category	P PROFESSIONAL CORE				
Course designed by	Department of Software Engineering				
Approval	32nd Academic Council Meeting,23rd July, 2016				

PURPOSE	To prepare software engineers to develop a secure software by incorporating security practices associated with information, systems and networks in all phases of the software development life cycle.

INSTRUCTIONAL OBJECTIVES		**STUDENT OUTCOMES**						
At the end of the course, student will be able to								
1.	Identify the various types of security attacks	a						
2.	Comprehend the various Encryption Techniques like Digital Signature , Hash Function	a	e					
3.	Familiarize the concept of security services and its practice in network	a	e					
4.	Practice the software security fundamentals	a	c					
5.	Understand how to build a secure software by incorporating security practices	a						

Session	Description of Topic	Contact hours	C-D-I-O	IOs	Reference
	UNIT 1-INTRODUCTION TO INFORMATION SECURITY	**7**			1,2
1.	An Overview of information security policies	1	C	1	1
2.	Types of threats-Security Attacks, Services and Security Mechanism for messages	2	C	1,2	1
3.	Information integrity Services, access control matrix.	2	D	1	1
4.	Cryptanalytic attack and non-Cryptanalytic attack	2	C,O	1,2	1,2
	UNIT II – SECURITY TECHNIQUES IN INFORMATION AND NETWORKS	**11**			
5.	AES- Confidentiality Using Symmetric Encryption	1	C,O	2	1
6.	Public Key Cryptography and RSA	1	C,D,O	3	1
7.	Key Management – Diffie Hellman Key Exchange, Elliptic Curve Arithmetic,ECC	2	C,O	2,3	1
8.	Message Authentication And Hash Function- Authentication Requirements, Authentication Functions- Message Authentication Code- Hash Function -Security of Hash Function and MAC.	3	C,O	2,3	1
9.	Digital Signature and Authentication Protocols- Digital Signature, Authentication Protocols, DSS	1	C,O	2,3	1,2
10.	Authentication Protocols- Kerberos, X.509 Authentication Service, Public Key Infrastructure	1	C,O	3	1,2
11.	**UNIT III - SECURITY SERVICES**	9			1,2
12.	Security At the Application Layer : E-mail, PGP and S/MIME	**1**	C,O	3	1,2
13.	Security at the Transport Layer : SSL and TLS- SSL Architecture-SSL Message Formats-Transport Layer Security- Four Protocols	3	C	3	1,2
14.	Security at the Network Layer: IPSEC- IP Security Overview- IP Security Architecture-Authentication Header-Encapsulating Security Payload-Combining Security Associations-Key management	3	C	3	1,2

15.	System Security- Intrusion Detection System- Firewalls-Construction and Working Principles	2	C	3	1,2
16.	Modern Encryption Techniques-Hashchain, Merkle trees-CASESTUDY	1	C	3	5,6
17.	**UNIT IV- SOFTWARE SECURITY FUNDAMENTALS**	9			3
18.	Introduction-Threats to software security-Benefits of Detecting Software Security	2	C	4	3
19.	Properties of Secure Software- Influencing the security properties of software- Defensive Perspective, Attacker's Perspective	2	C	4	3
20.	Building a Security Assurance Case, A Security Assurance Case Example, Incorporating Assurance Cases into the SDLC	3	C,D	4	3
21.	Requirements Engineering for secure software: SQUARE process Model, Requirements elicitation and prioritization	2	C	4	3
	UNIT V- ENGINEERING SECURE SOFTWARE	9			4
22.	Secure Software Architecture and Design: Software security practices for architecture and design: architectural risk analysis	1	C	5	3
23.	software security knowledge for architecture and design: security principles, security guidelines and attack patterns	1	C	5	3
24.	Secure Coding and Testing : Code analysis, Common Software Code Vulnerabilities, Source Code Review	1	C	5	3
25.	Coding Practices: Best Practices for Secure Coding	1	C	5	3
26.	Software Security testing: Contrasting Software Testing and Software Security Testing, Functional Testing, Risk-Based Testing, Penetration Testing	3	C	5	3
27.	Security testing considerations throughout the SDLC: Unit Testing, Testing Libraries and Executable Files, Integration Testing and System Testing	2	C	5	3
	Total Contact Hours	**45**			

LEARNING RESOURCES	
Sl. No.	**TEXT BOOKS**
1	William Stallings, " Cryptography and Network Security – Principles and Practices" Fourth Edition, Prentice Hall of India, 2006.
2	Beerhouse A.Forouzan and DebdeepMukhopadhyay ", Cryptography and Network Security" second edition,TataMCGraw Hill Publication, 2011.
3	Julia H. Allen, Gary McGraw "Software Security Engineering: A Guide for Project Managers" Addison-Wesley, 2008.
	REFERENCE BOOKS/OTHER READING MATERIAL
4	Gary McGraw "Software Security: Building Security In", Addison-Wesley, 2006.
5	https://www.emsec.rub.de/media/crypto/attachments/files/2011/04/becker_1.pdf, 2008
6	http://www.netsec.ethz.ch/publications/papers/light4.pdf

Course nature			**Theory**				
Assessment Method (Weightage 100%)							
In-semester	**Assessment tool**	Cycle test I	Cycle test II	Cycle Test III	Surprise Test	Quiz	**Total**
	Weightage	**10%**	**15%**	**15%**	**5%**	**5%**	**50%**
					End semester examination Weightage :		**50%**

15SE401M	**Multi Disciplinary Design**	**L**	**T**	**P**	**C**
		3	**0**	**0**	**3**

Co-requisite:	NIL
Prerequisite:	NIL
Data Book / Codes/Standards	NIL
Course Category	P | PROFESSIONAL CORE
Course designed by	Department of Software Engineering
Approval	32nd Academic Council Meeting, 23rd July 2016

PURPOSE	Students of any specialization at an undergraduate level learn courses related to various sub-domains (Multi-disciplinary) of their specialization individually. They are not exposed to understanding how the various multi-disciplinary fields interact and integrate in real life situations. It is very common that an expert in a particular domain models and designs systems or products oblivious of the impact of other subsystems. This lack of multi-disciplinary thinking is very blatantly visible when the students take up their major project during their final year. This course aims to develop appropriate skills on systemic thinking on how to identify and formulate a problem, decompose the problem into smaller elements, conceptualise the design, evaluate the conceptual design by using scientific, engineering and managerial tools, select, analyze and interpret the data, consideration of safety, socio-politico-cultural, risks and hazards, disposal, regional and national laws, costing and financial model and undertake documentation and finally presentation.

INSTRUCTIONAL OBJECTIVES		**STUDENT OUTCOMES**						
At the end of the course, student will be able								
1.	To subdivide a complex system into smaller disciplinary models, manage their interfaces and reintegrate them into an overall system model	a	c	e	f	i	l	
2.	To rationalize a system architecture or product design problem by selecting appropriate design variables, parameters and constraints	a	c	e	f	i	l	
3.	To design for value and quantitatively assess the expected lifecycle cost of a new system or product	a	c	e	f	i	l	
4.	To take on the challenges of teamwork, prepare a presentation in a professional manner, and document all aspects of design work.	a	c	e	f	i	l	

Session	**Description of Topic**	**Contact hours**	**C-D-I-O**	**IOs**	**Reference**
1	Introduction: Facilitating Multidisciplinary Projects		C,D,I,O	1,2,3,4	1,2
2	Identifying and formulating a problem				
3	System Modelling				
4	Thinking perspectives: Decomposition–Composition Thinking Hierarchical Thinking, Organizational Thinking, Life-Cycle Thinking, Safety Thinking, Risk Thinking, Socio-politico-cultural thinking, Environment thinking				
5	Decomposing a system – Identifying the major sub-systems				
6	Mathematical Modeling and Governing equations for each sub systems				
7	Objectives, Constraints and Design Variables				
8	Conceptual Design				
9	Collaborative Design – Disciplinary teams satisfy the local constraints while trying to match the global constraints set by the project coordinator.				
10	Tools for modeling, designing, analysis, data interpretation, decision making etc				
11	Design Analysis, evaluation and selection				
12	Costing and Financial model				
13	Documentation, reviewing and presentation				
	Total contact hours	**60**			

LEARNING RESOURCES	
Sl.	**REFERENCES**

No.	
1.	**Systems Design and Engineering: Facilitating Multidisciplinary Development Projects** G. Maarten Bonnema, Karel T. Veenvliet, Jan F. BroeninkDecember 15, 2015, CRC Press ISBN 9781498751261
2.	**Exploring Digital Design-Multi-Disciplinary Design Practices,** Ina Wagner , Tone Bratteteig , DagnyStuedahl, Springer-Verlag London, 2010, ISSN:1431-1496
	Additional references can be included by the respective departments based on the domain and / or theme.

Course nature						**Predominantly Practice complimented by theory**
Assessment Method (Weightage 100%)						
In-semester	**Assessment tool**	Review 1	Review 2	Review 3	Review 4	**Total**
	Weightage	**10%**	**25%**	**25%**	**40%**	**100%**
End semester examination Weightage :						**0%**

Pedagogy:

Theme or major/broad domains will be announced by the department every semester. Multi-disciplinary designs will be made by the students in groups (group size may be decided by the course coordinator), with the topic of interest falling within the theme or major/broad domains as announced by the department, applying any combinations of the disciplines in engineering. 3D modelling and / or simulation must be used to validate the design.

In a combination of lecture and hands-on experiences, students must be exposed to understand and analyse engineering designs (or products) and systems, their realization process and project management. Analysis of the design criteria for safety, ergonomics, environment, life cycle cost and sociological impact is to be covered. Periodic oral and written status reports are required. The course culminates in a comprehensive written report and oral presentation. If required guest lecturers from industry experts from the sub-domains may be arranged to provide an outside perspective and show how the system design is being handled by the industry. The Conceive Design Implement Operate (CDIO) principles must be taught to the students.

A full-scale fabrication is not within the purview /scope of this course. Of course this design, if scalable and approved by the department, can be extended as the major project work

This course is 100% internal continuous assessment.

15SE376L	**Minor Project II**	**L**	**T**	**P**	**C**
		0	**0**	**3**	**2**

Co-requisite:	NIL
Prerequisite:	NIL
Data Book / Codes/Standards	NIL
Course Category	P PROFESSIONAL
Course designed by	Department of Software Engineering
Approval	32nd Academic Council Meeting ,23rd July 2016

PURPOSE	To obtain an hands-on experience in converting a small novel idea / technique into a working model / prototype involving multi-disciplinary skills and / or knowledge and working in at team.							
INSTRUCTIONAL OBJECTIVES		**STUDENT OUTCOMES**						
At the end of the course, student will be able								
1.	To conceptualise a novel idea / technique into a product	c						
2.	To think in terms of multi-disciplinary environment		d					
3.	To understand the management techniques of implementing a project				k			
4.	To take on the challenges of teamwork, prepare a presentation in a professional manner, and document all aspects of design work.			g				

Session	Description of Topic	Contact hours	C-D-I-O	IOs	Reference
	An Multidisciplinary project to be taken up by a team of maximum of ten students. Development of prototype product, a 3D model, simulation, blueprint for a larger project and any other development work are permitted. The contribution of the individuals in the project should be clearly brought out. A combined report is to be submitted. A presentation is to be made for the reviewers on the work done by the candidate.		C,D,I	1,2,3,4	
	Total contact hours				

Course nature		**Project – 100% internal continuous assessment**	
Assessment Method (Weightage 100%)			
In-semester	**Assessment tool**	Refer the table	**Total**
	Weightage	Refer the table below	**100%**
End semester examination Weightage :			**0%**

Assessment components

Assessment component	Expected outcome	Evaluators	Criteria or basis	Marks
Project proposal (Review – I)	A short presentation to be delivered on: • A brief, descriptive project title (2-4 words). This is critical! • The 3 nearest competitors (existing solutions) and price. • Team members name, phone number, email, department/degree program, and year. • A description of the product opportunity that has been identified. To include: Documentation of the market need, shortcomings of existing competitive products, and definition of the target market and its size. • Proposed supervisor / guide	Panel of reviewers	Viability / feasibility of the project Extent of preliminary work done.	**0**

Review II	• Mission Statement / Techniques • Concept Sketches, Design Specifications / Modules & Techniques along with System architecture • Coding	Panel of reviewers	Originality, Multi-disciplinary component, clarity of idea and presentation, team work, handling Q&A.	**20**
Review III	• Final Concept and Model / Algorithm/ Technique • Drawings, Plans / programme output • Financial Model / costing • Prototype / Coding • Final Presentation and Demonstration	Panel of reviewers	Originality, Multi-disciplinary component, clarity of idea and presentation, team work, handling Q&A.	**50**
Final technical Report	A good technical report	Supervisor / Guide	Regularity, systematic progress, extent of work and quality of work	**30**
Total				**100**

15SE381L	Seminar II	L	T	P	C
		0	0	3	2
Co-requisite:	NIL				
Prerequisite:	NIL				
Data Book / Codes/Standards	NIL				
Course Category	P PROFESSIONAL				
Course designed by	Department of Software Engineering				
Approval	32nd Academic Council Meeting , 23rd July 2016				

PURPOSE	To inculcate the research culture among the students through literature reading, modelling a problem, analyzing and presenting.

INSTRUCTIONAL OBJECTIVES		**STUDENT OUTCOMES**						
At the end of the course, student will be able								
1.	To understand the research methodology adopted by various researchers	h	i	j				
2.	To mathematically model a problem, critically analyse it and adopt strategies to solve	b	c	e				
3.	To understand and present a well documented research	e	g					

Session	Description of Topic	Contact hours	C-D-I-O	IOs	Reference
	Guidelines for conducting 15SE381L Seminar for B.Tech 1. Upon registering for the course the student must identify a sub-domain of the degree specialization that is of interest to the student and start collecting research papers as many as possible. 2. After collecting sufficient number of research papers the student must peruse all the papers, meet the course faculty and discuss on the salient aspects of each and every paper. 3. The course faculty, after discussion with the student will approve TWO research papers that is appropriate for presentation. 4. The student must collect additional relevant reference materials to supplement and compliment the two research papers and start preparing the presentation. 5. Each student must present a 15-minute presentation on each of the approved research paper to the panel of evaluators. 6. The presenter must present one research paper within the first half of the semester (6 weeks) and another research paper in the next half of the semester (6 weeks) as per the schedule. 7. All other students registered for the course will form the audience. 8. The audience as well as the evaluators will probe the student with appropriate questions and solicit response from the presenter. 9. The presentation will be evaluated against 7 to 8 assessment criteria by 4 to 5 evaluators. 10. The score obtained through the presentations of TWO research papers will be converted to appropriate percentage of marks. This course is 100% internal continuous assessment.		C,D	1,2,3,4	
	Total contact hours	**30**			

Course nature		**100% internal continuous assessment.**		
Assessment Method (Weightage 100%)				
In-Semester	**Assessment tool**	Presentation 1	Presentation 2	**Total**
	Weightage	**50%**	**50%**	**100%**
End semester examination Weightage :				**0%**

EVALUATION OF SEMINAR PRESENTATIONS

Name of the Student: Date:

Register Number: Degree and Branch:

Topic:

Sl. No.	Criteria for Assessment	Evaluator 1	Evaluator 2	Evaluator 3	Evaluator 4	Evaluator 5
1	Understanding of the subject					
2	Clarity of presentation					
3	Appropriate use of Audio visual aids					
4	Whether cross references have been consulted					
5	Ability to respond to questions on the subject					
6	Time scheduling					
7	Completeness of preparation					

Poor	1	Below Average	2	Average	3	Good	4	Very Good	5

Overall Grades:

Remarks:

Signature of Course Coordinator

15SE386L	MOOCs II	L	T	P	C
		0	0	3	2
Co-requisite:	NIL				
Prerequisite:	NIL				
Data Book / Codes/Standards	NIL				
Course Category	P PROFESSIONAL				
Course designed by	Department of Software Engineering				
Approval	32nd Academic Council Meeting, 23rd July 2016				

PURPOSE	To offer students the opportunity to study with the world's best universities by integrating select MOOCs in a regular degree programme and providing students full credit transfer, as per university regulations, if they earn a "Verified / Completion Certificate" and take a proctored examination through a secure, physical testing center.

INSTRUCTIONAL OBJECTIVES		**STUDENT OUTCOMES**						
At the end of the course, student will be able								
1.	To apply the concepts, theories, laws, technologies learnt herein to provide engineering solutions.	f	h	i	j			

Course nature		**Online - 100% internal continuous assessment.**				
Assessment Method (Weightage 100%)						
In-semester	**Assessment tool**	Quiz	Assignment	Non-proctored / Unsupervised Tests	Proctored / Supervised Test	**Total**
	Weightage	**25%**	**25%**	**10%**	**40%**	**100%**
End semester examination Weightage :						**0%**

Registration process, Assessment and Credit Transfer:

1. Students can register for courses offered by approved global MOOCs platforms like edX, Coursera or Universities with which SRM partners specifically for MOOCs.
2. Annually, each department must officially announce, to the students as well as to the Controller of Examinations, the list of courses that will be recognized and accepted for credit transfer.
3. The department must also officially announce / appoint one or more faculty coordinator(s) for advising the students attached to them, monitoring their progress and assist the department in proctoring the tests, uploading the marks / grades, and collecting and submitting the graded certificate(s) to the CoE, within the stipulated timeframe.
4. Student who desires to pursue a course, from the above department-approved list, through MOOCs must register for that course during the course registration process of the Faculty of Engineering and Technology, SRM University.
5. The maximum credit limits for course registration at SRM will include the MOOCs course registered.
6. The student must periodically submit the marks / grades obtained in various quizzes, assignments, tests etc immediately to the Faculty Advisor or the Course Coordinator for uploading in the university's academic module.
7. The student must take the final test as a Proctored / Supervised test in the university campus.
8. The student must submit the "Certificate of Completion" as well as the final overall Marks and / or Grade within the stipulated time for effecting the grade conversion and credit transfer, as per the regulations. It is solely the responsibility of the individual student to fulfil the above conditions to earn the credits.
9. The attendance for this course, for the purpose of awarding attendance grade, will be considered 100% , if the credits are transferred, after satisfying the above (1) to (7) norms; else if the credits are not transferred or transferable, the attendance will be considered as ZERO.

15SE491L	Industrial Module II	L	T	P	C
		0	0	3	2
Co-requisite:	NIL				
Prerequisite:	NIL				
Data Book / Codes/Standards	NIL				
Course Category	P PROFESSIONAL				
Course designed by	Department of Software Engineering				
Approval	32nd Academic Council Meeting , 23rd July 2016				

PURPOSE	To offer students the opportunity to interact with industries and learn the best practices adopted by them.

INSTRUCTIONAL OBJECTIVES		**STUDENT OUTCOMES**						
At the end of the course, student will be able								
1.	To obtain an insight into the current industrial trends and practices	h						
2.	To obtain an insight into the technologies adopted by industries	k						
3.	To obtain an insight into the technical problems encountered by the industries and the scope for providing solutions.	e						
4.	To network with industry	h						

Description of Topic	**Contact hours**	**C-D-I-O**	**IOs**	**Reference**
1. The department will identify and shortlist few emerging topics that are trending in industry. 2. The department will identify experts from industry who are willing to deliver modules on the shortlisted topics. 3. The identified expert will assist the department in formulating the course content to be delivered as a 30-hour module, prepare lectures notes, ppt, handouts and other learning materials. 4. The department will arrange to get the necessary approvals for offering the course, from the university's statutory academic bodies well before the actual offering. 5. The department must officially announce, to the students as well as to the Controller of Examinations, the list of courses that will be offered as industry module. 6. The department must also officially announce / appoint one or more faculty coordinator(s) for advising the students attached to them, monitoring their progress and assist the department in proctoring/supervising/assessment the quizzes, assignments, tests etc, uploading the marks, attendance etc, within the stipulated timeframe. 7. The Student who desires to pursue a course, from the above department-approved list, must register for that course during the course registration process of the Faculty of Engineering and Technology, SRM University. 8. The maximum credit limits for course registration at SRM will include the Industry Module also. 9. All academic requirements of a professional course like minimum attendance, assessment methods, discipline etc will be applicable for this Industry Module. 10. The course will be conducted on weekends or beyond the college regular working hours.		C,D,I,O	1,2,3,4	
Total contact hours	**30**			

Course nature		**100% internal continuous assessment.**					
Assessment Method – Theory Component (Weightage 50%)							
In-semester	Assessment tool	Cycle test I	Cycle test II	Cycle Test III	Surprise Test	Quiz	Total
	Weightage	10%	15%	15%	5%	5%	50%
End semester examination Weightage							50%

15SE496L	Major Project	L	T	P	C
		0	0	24	12
Co-requisite:	NIL				
Prerequisite:	NIL				
Data Book / Codes/Standards	NIL				
Course Category	P PROFESSIONAL CORE				
Course designed by	Department of Software Engineering				
Approval	32nd Academic Council Meeting ,23rd July 2016				

PURPOSE	The Major Project experience is the culminating academic endeavor of students who earn a degree in their Undergraduate Programs. The project provides students with the opportunity to explore a problem or issue of particular personal or professional interest and to address that problem or issue through focused study and applied research under the direction of a faculty member. The project demonstrates the student's ability to synthesize and apply the knowledge and skills acquired in his/her academic program to real-world issues and problems. This final project affirms students' ability to think critically and creatively, to solve practical problems, to make reasoned and ethical decisions, and to communicate effectively.

INSTRUCTIONAL OBJECTIVES		**STUDENT OUTCOMES**						
At the end of the course, student will be able								
1.	To provide students with the opportunity to apply the knowledge and skills acquired in their courses to a specific problem or issue.	a	c		e	f		i
2.	To allow students to extend their academic experience into areas of personal interest, working with new ideas, issues, organizations, and individuals.	a	c		e	f		i
3.	To encourage students to think critically and creatively about academic, professional, or social issues and to further develop their analytical and ethical leadership skills necessary to address and help solve these issues.	a	c		e	f	h	i
4.	To provide students with the opportunity to refine research skills and demonstrate their proficiency in written and/or oral communication skills.	a	c		e	f	g	i
5.	To take on the challenges of teamwork, prepare a presentation in a professional manner, and document all aspects of design work.			d			g	

Session	**Description of Topic**	**Contact hours**	**C-D-I-O**	**IOs**	**Reference**
	1. The Major project is a major component of our engineering curriculum: it is the culmination of the program of study enabling the students to showcase the knowledge and the skills they have acquired during the previous four years, design a product/service of significance, and solve an open-ended problem in engineering. 2. Each student must register to the project course related to his or her program 3. Major Project course consists of one semester and would be allowed to register only during the final year of study. 4. The Major Project may be initiated during the pre-final semester but will be assessed and credits transferred only during the last semester of study, upon completion of all other degree requirements. Generally the undergraduate major project is a team based one. 5. Each team in the major project course will consist of maximum of 5 students. 6. Each project will be assigned a faculty, who will act as the supervisor. 7. The project shall be driven by realistic constraints like that related to economic, environmental, social, political, ethical, health & safety, manufacturability and sustainability.		C,D,I,O	1,2,3,4,5	

	8. Each group must document and implement a management structure. Group leadership roles must be clearly identified including who has responsibility for monitoring project deliverables and group coordination.				
	9. A group project may be interdisciplinary, with students enrolled in different engineering degrees, or in Engineering plus other faculties such as Management, Medical and Health Sciences, Science and Humanities. 10. Each student team is expected to maintain a log book that would normally be used to serve as a record of the way in which the project progressed during the course of the session. 11. Salient points discussed at meetings with the supervisor (i.e., suggestions for further meetings, changes to experimental procedures) should be recorded by the student in order to provide a basis for subsequent work. 12. The logbook may be formally assessed; 13. The contribution of each individual team member will be clearly identified and the weightage of this component will be explicitly considered while assessing the work done. 14. A project report is to be submitted on the topic which will be evaluated during the final review. 15. Assessment components will be as spelt out in the regulations. 16. The department will announce a marking scheme for awarding marks for the different sections of the report. 17. The project report must possess substantial technical depth and require the students to exercise analytical, evaluation and design skills at the appropriate level.				
	Total contact hours				

Course nature		**Project – 100 % Internal continuous Assessment**			
Assessment Method (Weightage 100%)					
In-semester	**Assessment tool**	Review 1	Review 2	**Review 3**	**Total**
	Weightage	**10%**	**15%**	**20%**	**45%**
End semester examination	**Assessment Tool**	**Project Report**	**Viva Voce**		
	Weightage :	**25%**	**30%**		**55%**

15SE321E	Principles Of Programming Languages	L	T	P	C
		3	0	0	3
Co-requisite:	NIL				
Prerequisite:	NIL				
Data Book / Codes/Standards	NIL				
Course Category	P PROFESSIONAL ELECTIVE				
Course designed by	Department of Software Engineering				
Approval	32nd Academic Council Meeting, 23rd July, 2016				

PURPOSE	The main purpose of this course is to describe syntax and semantics of programming languages.							
INSTRUCTIONAL OBJECTIVES		**STUDENT OUTCOMES**						
At the end of the course, student will be able to								
1.	Appreciate the basic and advanced feature of language built ins	i						
2.	Handle syntax and semantics of programming languages	b						
3.	Distinguish data, data types, and basic statements	i						
4.	Observe call-return architecture and ways of implementing them	b						
5.	Outline object-orientation, concurrency, and event handling in programming languages	k						

Session	Description of Topic	Contact hours	C-D-I-O	IOs	Reference
	UNIT I: Introduction to Programming Language	9			
1.	Concepts of Programming Languages	1	C,D	1	1
2.	Programming Domains	1	C	1	1
3.	Language evaluation Criteria-Language Categories	1	C	1	1
4.	Evolution of programming languages	1	C	1	1
5.	IBM 704 and Fortran	1	C,D	1	1
6.	Functional Programming LISP	1	C	1	1
7.	ALGOL60-COBOL-BASIC	1	C,D	1	1
8.	Prolog- C++ - Java - C#	2	C,D	1	1,5
	UNIT II: SYNTAX AND SEMANTICS	9			
9.	Describing syntax	1	C	2	1,2,5
10.	Context-free grammars ,attribute grammars	1	C	2	1,2,5
11.	Describing semantics	1	C	2	1,2,5
12.	Lexical analysis – parsing	2	C,I	2	1,2
13.	Recursive-decent	2	C,I	2	1
14.	Bottom-up parsing	2	C,D,I	2	1
	UNIT III: DATA TYPES, AND BASIC STATEMENTS	**9**			
15.	Names – variables – binding	2	C	3	1,3
16.	Scope – scope rules	1	C	3	1
17.	Primitive data types – strings – array types-record types-union types	2	C	3	1,3,4
18.	Arithmetic expressions – overloaded operators – type conversions – relational and boolean expressions	2	C,D,I	1,3	1,3
19.	Control structures – selection – iterations – branching statements	2	C,D	1,3	1,3
	UNIT IV: SUBPROGRAMS AND IMPLEMENTATIONS	**9**			
20.	Subprograms – design issues – local referencing	1	C	4	1
21.	Parameter passing – overloaded methods – generic methods	2	C,I	4	1
22.	design issues for functions	3	C,D,I	4	1
23.	Implementing simple subprograms – stack and dynamic local variables	2	C,I	4	1

24.	Nested subprograms – blocks – dynamic scoping	1	C,I	4	1
	UNIT V: OBJECT-ORIENTATION, CONCURRENCY, AND EVENT HANDLING	**9**			
25.	Object-orientation – design issues for OOP languages	1	C	3,5	1
26.	Implementation of object-oriented constructs	2	C,I	3,5	1
27.	Concurrency – semaphores – monitors	2	C,D,I	5	1
28.	Message passing – threads	2	C	3,5	1
29.	Exception handling-event handling	2	C	3,5	1
Total contact hours		45			

LEARNING RESOURCES	
Sl. No.	**TEXT BOOKS**
1.	Robert W. Sebesta, "Concepts of Programming Languages", Tenth Edition, Addison Wesley, 2012.
	REFERENCE BOOKS/OTHER READING MATERIAL
2.	Michael L. Scott, "Programming Language Pragmatics", Third Edition, Morgan Kaufmann, 2013.
3.	"The Scheme programming language", Fourth Edition, MIT Press, 2009.
4.	Jeffrey D. Ullman, "Elements of ML programming", Second Edition, Prentice Hall, 1998
5.	Richard A. O'Keefe, "The craft of Prolog", MIT Press, 2009.

Course nature				**Theory**			
Assessment Method (Weightage 100%)							
In-semester	**Assessment tool**	Cycle test I	Cycle test II	Cycle Test III	Surprise Test	Quiz	**Total**
	Weightage	**10%**	**15%**	**15%**	**5%**	**5%**	**50%**
					End semester examination Weightage :		**50%**

15SE322E	E-Commerce	L	T	P	C
		3	0	0	3
Co-requisite:	NIL				
Prerequisite:	NIL				
Data Book / Codes/Standards	NIL				
Course Category	P PROFESSIONAL ELECTIVE				
Course designed by	Department of Software Engineering				
Approval	32nd Academic Council Meeting, 23rd July, 2016				

PURPOSE	Big corporations and financial institutions use the internet to exchange the financial data to facilitate domestic and international business. Data integrity and security are very hot and pressing issues for Electronic commerce. This course provides better understanding of the technical aspects and process of E-commerce.

INSTRUCTIONAL OBJECTIVES		**STUDENT OUTCOMES**						
At the end of the course, student will be able to								
1.	Distinguish the E-Commerce framework and business model applications	a	g	h				
2.	Outline the Infrastructure of E-commerce	g						
3.	Apply security algorithms	e	k					
4.	Identify and operate e-payment mechanisms.	i						

Session	Description of Topic	Contact hours	C-D-I-O	IOs	Reference
	UNIT I: INTRODUCTION	7			
1.	Introduction to E- Commerce	1	C	1	1
2.	Generic Framework of E- Commerce	1	C	1	1
3.	Business Models	2	C	1	1
4.	Consumer Oriented E- Commerce Applications	2	C	1	1
5.	Mercantile Process Models	1	C	1	1
	UNIT II: NETWORK INFRASTRUCTURE AND MOBILE COMMERCE	9			
6.	Network Infrastructure for E-Commerce	1	C	1,2	1,3
7.	Market forces behind I Way, Component of I way Access Equipment	2	C	1,2	1,3
8.	Global Information Distribution Network	1	C	2	2
9.	Broad band Telecommunication (ATM, ISDN, Frame Relay).	2	C	2	2
10.	Mobile Commerce, Mobile Computing Application	2	C	2	2
11.	Wireless Application Protocols, WAP Technology	1	C,D	2	2
	UNIT III: WEB SECURITY	**10**			
12.	Security Issues on Web- World Wide Web & Security	2	C,D	2,3	1
13.	Importance of Firewall- Components of Firewall,	2	C,I	3	2
14.	Factors to consider in Firewall Design- Limitations of Firewalls	2	C,D	2,3	2
15.	Transaction Security- Client Server Network	2	C	3	2
16.	Emerging Client Server Security Threats-Network Security.	2	C	3	2
	UNIT IV: SECURITY	**10**			
17.	Encryption Techniques	1	C	3	2
18.	Symmetric Encryption- Keys and Data Encryption standard, Triple encryption,	3	C,D	3	2
19.	Asymmetric encryption- Secret Key Encryption, Public and Private pair key encryption	3	C,D	3	2
20.	Digital Signatures-Virtual Private Network (VPN)	3		2,3	2

			C,D		
	UNIT V: ELECTRONIC PAYMENTS	**9**			
21.	Overview of Electronics payments, The SET Protocol, Payment Gateway, Certificates	2	C	4	1,4
22.	Digital Token, Smart Cards, Credit Cards, Magnetic Strip Cards, E-Checks, Credit/ Debit card EPS	2	C,D	4	1,4
23.	Mobile Payments, Online Banking, Home banking, Emerging financial Instruments,	2	C,D	4	1,4
24.	EDI Application in Business, E-commerce laws, Forms of Agreement, Government Policies and Agenda	2	C,D	4	2,5
25.	E-Commerce Strategy in Business Models and Internet Start-ups: A Business Case Study.	1	O	4	2,5
	Total contact hours	45			

LEARNING RESOURCES	
Sl. No.	**TEXT BOOKS**
1.	Ravi Kalakota and Andrew B Whinston, "Frontiers of Electronic Commerce", Pearson Education, 2013.
2.	Greenstein and Feinman, "E-Commerce", TMH,2001
	REFERENCE BOOKS/OTHER READING MATERIAL
3.	Denieal Amor, "The E-Business Revolution", Addison Wesley, Second edition 2002.
4.	Bajaj & Nag, "E-Commerce: The Cutting Edge of Business", TMH,Second Edition 2005
5.	DiwanParag / Sharma Sunil , "E-commerce :A Manager's Guide to E-Business"First edition 2000

Course nature		**Theory**					
Assessment Method (Weightage 100%)							
In-semester	**Assessment tool**	Cycle test I	Cycle test II	Cycle Test III	Surprise Test	Quiz	**Total**
	Weightage	**10%**	**15%**	**15%**	**5%**	**5%**	**50%**
					End semester examination Weightage :		**50%**

15SE323E	Design Patterns	L	T	P	C
		3	0	0	3
Co-requisite:	NIL				
Prerequisite:	NIL				
Data Book / Codes/Standards	NIL				
Course Category	P PROFESSIONAL ELECTIVE				
Course designed by	Department of Software Engineering				
Approval	32nd Academic Council Meeting, 23rd July, 2016				

PURPOSE	To learn how to reuse the design solution for different kind of software problems and to understand about the interactions between objects.

INSTRUCTIONAL OBJECTIVES		**STUDENT OUTCOMES**						
At the end of the course, student will be able to								
1.	Conceive the importance of reuse of solution for common problems in software development.	c	a					
2.	Identify the appropriate patterns for design problems.	a						
3.	Implement the various design pattern solution for appropriate scenarios	e						
4.	Refactoring the badly designed program properly using patterns.	b	a	c				

Session	Description of Topic	Contact hours	C-D-I-O	IOs	Reference
	UNIT I: OBJECT DESIGN	**9**			
1.	Basics of UML - Class Diagram - Interaction Diagrams – relationships	2	C	1	1
2.	Object design - Reuse concepts – Solution objects, Inheritance, and design patterns	2	C	1	1
3.	Principles and Strategies of Design Patterns - Open-Closed Principle – Designing from Context - Encapsulating Variation-Abstract Classes vs Interfaces	3	C	2	2
4.	Design Patterns & Architecture, Gang of Four (GoF) patterns	2	C,D	2	2
	UNIT II : BEHAVIORAL PATTERNS (STRATEGY, MEDIATOR, TEMPLATE METHOD)	**9**			
5.	Strategy pattern- Intent, Motivation, Applicability, Structure, Participants, collaborations, consequences	1	C	2,3	2,3
6.	Strategy pattern- Implementation and sample code	1	C,I	2,3	2,4
7.	Mediator - Intent, Motivation, Applicability, Structure, Participants, collaborations, consequences	1	C	2,3	2,3
8.	Mediator- Implementation and sample code	1	C,I	2,3	2,3,4
9.	Template Method- Intent, Motivation, Applicability, Structure, Participants, collaborations, consequences	1	C	2,3	2,3
10.	Template Method- Implementation and sample code	2	C,I	2,3	2,4
11.	Case study: Identify which pattern is applicable for the given case study and justify	2			
	UNIT III : CREATIONAL PATTERNS (ABSTRACT FACTORY PATTERN, FACTORY METHOD PATTERN, SINGLETON PATTERN)	**9**			
12.	Abstract factory- Intent, Motivation, Applicability, Structure, Participants, collaborations, consequences	1	C	2,3	2,3
13.	Abstract factory- Implementation and sample code	1	C,I	2,3	2,4
14.	Singleton pattern- Intent, Motivation, Applicability, Structure, Participants, collaborations, consequences	1	C	2,3	2,3

15.	Singleton pattern Implementation and sample code	1	C,I	2,3	2,4
16.	Factory method pattern- Intent, Motivation, Applicability, Structure, Participants, collaborations, consequences	1	C	2,3	2,3
17.	Factory method pattern Implementation and sample code	2	C,I	2,3	2,4
18.	Case study: Identify which pattern is applicable for the given case study and justify	2			
	UNIT IV : STRUCTURAL PATTERNS (ADAPTER, BRIDGE, DECORATOR, FACADE)	**9**			
19.	Adapter- Intent, Motivation, Applicability, Structure, Participants, collaborations, consequences	1	C	2,3	2,3
20.	Adapter- Implementation and sample code	1	C,I	2,3	2,3
21.	Bridge- Intent, Motivation, Applicability, Structure, Participants, collaborations, consequences	1	C	2,3	2,3
22.	Bridge- Implementation and sample code	1	C,I	2,3	2,4
23.	Decorator, Facade- Intent, Motivation, Applicability, Participants, collaborations, consequences Structure	1	C	2,3	2,3
24.	Decorator, Facade- Implementation and sample code	2	C,I	2,3	2,4
25.	Case study: Identify which pattern is applicable for the given case study and justify	2			
	UNIT V : CASE STUDY (REVERSE ENGINEERING)	**9**			
26.	What to Expect from Design Patterns	1	C	1	2
27.	The Pattern Community An Invitation	1	C	1	2
28.	A Parting Thought	1	C	2,4	2
29.	A Case Study : Designing a Document Editor : Design Problems, Document Structure	2	C	2,4	2
30.	Formatting, Embellishing the User Interface, Supporting Multiple Look-and-Feel Standards	2	C	2,4	2
31.	Supporting Multiple Window Systems, User Operations Spelling Checking and Hyphenation	2	C	2,4	2
	Total contact hours	45			

LEARNING RESOURCES	
Sl.No.	**TEXT BOOKS**
1.	Bruegge, Bernd and Allen H. Dutoit. “Object-Oriented Software Engineering: UsingUML, Patterns and Java”,Pearson: Prentice Hall Publishers 2004
2.	Erich Gamma, Richard Helm, “Design Patterns: Elements of reusable software development”, Pearson Education, 2005
	REFERENCE BOOKS/OTHER READING MATERIAL
3.	Alan Shalloway, James R Trott "Design pattern explained", Pearson Education, 2005.
4.	Eric Freeman, Elisabeth Robson, Bert Bates, and Kathy Sierra, “Head First Design Patterns”, O’reillyPublications, 2004.

Course nature		**Theory**					
Assessment Method (Weightage 100%)							
In-semester	**Assessment tool**	Cycle test I	Cycle test II	Cycle Test III	Surprise Test	Quiz	**Total**
	Weightage	**10%**	**15%**	**15%**	**5%**	**5%**	**50%**
					End semester examination Weightage :		**50%**

15SE324E	**Multimedia Systems**	**L**	**T**	**P**	**C**

		3	**0**	**0**	**3**
Co-requisite:	NIL				
Prerequisite:	NIL				
Data Book / Codes/Standards	NIL				
Course Category	P PROFESSIONAL ELECTIVE				
Course designed by	Department of Software Engineering				
Approval	32nd Academic Council Meeting, 23rd July, 2016				

PURPOSE	Students in this course will learn a broad range of digital media skills, concepts, terminologies, formats, trends, and infrastructure requirements. Through techniques for producing and distributing digital images, audio, video, presentations and web sites. Students will leave this class with a website portfolio that showcases their digital media skills and projects.

INSTRUCTIONAL OBJECTIVES		**STUDENT OUTCOMES**						
At the end of the course, student will be able to								
1.	Identify various properties of audio, image, video and animation data and how different they are from text**.**	c	e					
2.	Describe some compression techniques of various media types	c	e					
3.	Create and Manipulate Animation and multimedia presentation.	c						
4.	Elaborate and compare on the history of multimedia techniques	c						
5.	Describe basics and latest advances in multimedia	c						

Session	**Description of Topic**	**Contact hours**	**C-D-I-O**	**IOs**	**Reference**
	UNIT I: Introduction to Multi Media	**9**			
1.	Multimedia an Overview, Aspects of Multimedia	1	C	1	1
2.	Uses of Multimedia	1	D	1	1
3.	Digital representation	1	C,D	1	1
4.	Visual display system(LCD VS LED)	1	C	1	1
5.	Multimedia I/O Technologies MIDI	1	C,O	3	1
6.	Image Scanner	1	C,D,O	1	1
7.	Digital Voice and Audio – MIDI	1	C,O	1	1,6
8.	Video Still Image Processing	1	C,O	1	1
9.	Full motion Video controllers	1	C,O	1	1
	UNIT II: Text and Digital Imaging	**9**			
10.	Text – Types of Text	1	C,D	1	1
11.	Text Compression	1	C,O	2	1
12.	Image and its Types- Color Models	1	C,D	2	1
13.	Image Acquisition, Color Depth	1	C,I,O	1	1
14.	Layers and Layer Options	1	C,O	1,3	1
15.	Cropping and Colorizing Effects	1	C,O	1,3	1
16.	Dithering and Interlacing	1	C,O	1,3	1
17.	Text and Anti-Aliasing	1	C,O	1,3	1
18.	Image Formats	1	C,O	1,2	2
	UNIT III: **DIGITAL AUDIO**	**9**			
19.	Fundamental Characteristics of Sound	1	C,D	1,3,5	1
20.	Elements of Audio System	1	C,D	1,3	1
21.	Audio Mixer	1	C,O	1,3	1
22.	Digital Audio using a tool	1	C,I	1,3,4	1
23.	Sound Card Configuration	1	C,O	1,4,5	1,2
24.	Audio Recording, Sample rate and Bit-Depth	1	C,O	1,4,5	2
25.	Audio Editing- Dubbing and Mixing	1	C,O	1,4,5	2
26.	Audio File Formats- Codecs and compression options	1	C	1,2,4	2
27.	Cross Platform Compatibility	1	C	1,4	2
	UNIT IV:**(Windows Movie Maker)**	**9**			
28.	Digitizing analog footage via capture cards	1	D,O	3,4,5	4,5
29.	Transferring digital footage via FireWire	1	C,O	4	4,6

30.	Editing, mixing, scrubbing, and applying transitions and effects	1	C,D,O	4,5	4
31.	Titles and overlays	1	C,O	4,5	4
32.	Adding narration and soundtracks	1	C,O	3,4,5	4
33.	Frame rates, resolution, image size and color depth	1	O	3,4,5	4
34.	Video file formats, codecs and compression options	1	C,O	2,3,4	4,5
35.	Distribution options: Internet vs. CD/DVD vs. Videotape	1	C	4,5	4,5
36.	Case Study using Movie Making tool	1	O	3,4,5	4
	UNIT V: **Animation Technologies**				
37.	Advanced Multimedia- Virtual Reality & Augmented Reality	1	D,O	3,5	2,3,7
38.	Exploring the Animation tool Interface	1	C,O	1,3	3,7
39.	Learning Transforms in the Animation tool	1	C,O	3,5	3,7
40.	NURBS Curves and Surfaces	1	C,O	3,5	3,7
41.	Working with NURBS-Spline Coordinate	1	C,D	3,5	3,7
42.	Sample animations through the tool	1	C,D,O	3,5	3,7
43.	Polygon Primitives	1	C,D	3,5	3,7
44.	Editing Polygon Surfaces	1	C,D,O	3,4,5	3,7
45.	Subdivision surfaces using Animation tool	1	C,O	3,4,5	3,7
	Total contact hours	45			

LEARNING RESOURCES	
Sl. No.	**TEXT BOOKS**
1.	Ranjan Parekh, *"Principles Of Multimedia",* The McGraw - Hills Company, Twelfth Reprint 2011.
2.	Ralf Steinmetz, KlaraNahrstedt, *"Multimedia System",*X-media-Publishing, 2014.
3.	Tickoo, " Autodesk Maya 2011-Comprehensive Guide" Pearson- CADCIM Technologies
	REFERENCE BOOKS/OTHER READING MATERIAL
4.	Tim Morris, *"Multimedia Systems:Delivering, Generating and Interacting with Multimedia"***,**Springer, 2012
5.	Peter Wells, "A Beginners guide to Digital Video", AVA Publishing –Switzerland, 2004.
6.	Bruce Wands, "Digital Creativity- Digital Media and the Internet"- John wiley publication, 2002.
	WEB REFERENCE
7.	http://www.cadcimtech.com/maya_2011/maya_2011.htm

Course nature				**Theory**			
Assessment Method (Weightage 100%)							
In-semester	**Assessment tool**	Cycle test I	Cycle test II	Cycle Test III	Surprise Test/ Quiz	**Assignment**	**Total**
	Weightage	**10%**	**15%**	**15%**	**5%**	**5%**	**50%**
						End semester examination Weightage :	**50%**

15SE325E	System Software	L	T	P	C
		3	0	0	3
Co-requisite:	NIL				
Prerequisite:	NIL				
Data Book / Codes/Standards	NIL				
Course Category	P PROFESSIONAL ELECTIVE				
Course designed by	Department of Software Engineering				
Approval	32nd Academic Council Meeting, 23rd July, 2016				

PURPOSE	To have an understanding of foundations of design of assemblers, loaders, linkers, macro-processors compilers, operating system and system software tools.

INSTRUCTIONAL OBJECTIVES		**STUDENT OUTCOMES**						
At the end of the course, student will be able to								
1.	Understand the relationship between system software architecture and machine and able to trace the path of a source code to object code and to executable file	b	d					
2.	Ability to design and implementation of macro processors, linkers and loaders	b	d					
3.	Ability to analyze the functions of compilers	b	e					
4.	Ability to understand the concepts of operating system	a						
5.	Ability to understand about system software tools	b	k					

Session	Description of Topic	Contact hours	C-D -I-O	IOs	Reference
	UNIT I: ASSEMBLER	**10**			
1.	Introduction-Systems Programming- Language Translators	1	C	1	1,4
2.	System software and SIC/XE machine architecture	1	C	1	1,4
3.	Basic assembler functions	1	C	1	1,4
4.	Machine dependent assembler features	2	C	1	1,4
5.	Machine independent assembler features	2	C	1	1,4
6.	Assembler Design Options-One pass assemblers	1	C,I	1	1,4
7.	Multi pass assemblers	1	C,I	1	1,4
8.	Implementation example -MASM Assembler	1	C,I	1	1,4
	UNIT II: MACROPROCESSOR AND LOADER	**12**			
9.	Basic Macro processor functions	1	C	1,2	1
10.	Concept of Parameters in Macros	1	C	1,2	1
11.	Conditions in Macro Definitions	1	C	1,2	1
12.	Macro Programming Techniques	1	C,I	1,2	1
13.	Design Issues of Macro Processors	1	C,I	1,2	1
14.	Principles of Loading Operation	1	C	1,2	1,5
15.	Different Loading Schemes	2	C	1,2	1,5
16.	Principles of Linking- Relocation and Code Modification	1	C	1,2	1,5
17.	Linking Methods	1	C	1,2	1,5
18.	Design of Linkers	1	C,I	1,2	1,5
19.	Implementation Example-MSDOS Linker	1	C,I	1,2	1,5
	UNIT III:COMPILER	**10**			
20.	Compilers-Variants of Compilers	1	C	1,3	1
21.	Structured Language- Regular Expressions-Finite Automata	1	C	1,3	1

22.	Grammars	2	C	1,3	1
23.	Lexical Analysis	1	C	1,3	1
24.	Syntax Analysis	1	C	1,3	1
25.	Semantic Analysis-Parameter Passing	1	C	1,3	1
26.	Code Generation	1	C	1,3	1
27.	Code Optimization	1	C	1,3	1
28.	Implementation Example-Java compiler and Environment	1	C,I	1,3	1
	UNIT IV: CONCEPTS OF OPERATING SYSTEM	**7**			
29.	The Scope and Functions of Operating Systems	1	C	4	1,3
30.	Memory Management Schemes	2	C	4	1,3
31.	Interrupt	1	C	4	1,3
32.	Operating System Design options	2	C	4	1,2
33.	Implementation Example-Windows 10	1	C,I	4	1
	UNIT V: SYSTEM TOOLS	**6**			
34.	Editors-Introduction- Document Editor	1	C	5	1
35.	The Design of a Text Editor, Sound Editor	1	C,I	5	1
36.	Case Studies	1	C	5	1
37.	Debuggers- Introduction-Types of Errors	1	C	5	1
38.	Debugging Procedures- Classification of Debuggers	1	C	5	1
39.	Debugging Using Hardware-Use of Debugger in Unix C	1	C,I	5	1
	Total contact hours	**45**			

LEARNING RESOURCES	
Sl. No.	**TEXT BOOKS**
1	Srimanta Pal, "*Systems Programming"*, Oxford University Press,2011.
	REFERENCE BOOKS/OTHER READING MATERIAL
2	Leland L. Beck and D.Manjula, "*System Software – An Introduction to Systems Programming*", 3rd Edition, Pearson Education Asia, 2009.
3	D. M. Dhamdhere, "*Systems Programming and Operating Systems*", Second Revised Edition, Tata McGraw-Hill, 2000.
4	John J. Donovan "*Systems Programming*", Tata McGraw-Hill Edition, 2000.
5	John R. Levine," Linkers & Loaders" – Harcourt India Pvt. Ltd., Morgan Kaufmann Publishers, 2000.

Course nature				**Theory**			
Assessment Method (Weightage 100%)							
In-semester	**Assessment tool**	Cycle test I	Cycle test II	Cycle Test III	Surprise Test	Quiz	**Total**
	Weightage	**10%**	**15%**	**15%**	**5%**	**5%**	**50%**
					End semester examination Weightage :		**50%**

15SE326E	Principles Of Compiler Design	L	T	P	C
		3	0	0	3
Co-requisite:	NIL				
Prerequisite:	NIL				
Data Book / Codes/Standards	NIL				
Course Category	P PROFESSIONAL ELECTIVE				
Course designed by	Department of Software Engineering				
Approval	32nd Academic Council Meeting ,23rd July, 2016				

PURPOSE	To acquire knowledge in design aspects of a compiler and master all the phases of compile and to design and implement a programming language.

INSTRUCTIONAL OBJECTIVES		**STUDENT OUTCOMES**						
At the end of the course, student will be able to								
1.	Describe and Analyze the structures and parse using grammars	a	b					
2.	Compose components like parser, analyzers and code generators	a	e	k				
3.	Optimize the code for construction.	a	e	k				
4.	Familiarize with the concept of computability	a						
5.	Expose to the concept of compiler construction.	a	e	k				

Session	Description of Topic	Contact hours	C-D-I-O	IOs	Reference
	UNIT 1 : Language Translation and Execution	**09**			
1.	**Introduction** - Language Processors – Structure of Compiler, Translation and Interpretation,	2	C	1	1-6
2.	The Tasks of a Compiler ,Data Management in a Compiler and Types of Compiler	2	C	1	1,3,4,6
3.	**Lexical Analysis** – Role of Lexical Analyzer, Input Buffering-Specification & Recognition of Tokens	2	C,D	1	1,3,4,6
4.	Regular Grammars and Finite Automata	3	C,D,I	1	1,3,4,6
	UNIT II: Syntax Analysis and Compiler Semantic Analysis	**09**			
5.	**Syntax Analysis** – Context-Free Grammars Top-Down Analysis and LL(1) Parsers	3	C,D,I	2	1,3,4,6
6.	Bottom-Up Analysis and LR Parsers	3	C,D,I	2	1,3
7.	Attribute Grammars –Basic Concepts	2	C,D	2	1,3
8.	**Semantic Analysis –** Type Checking	1	C		1,2
	UNIT III: Intermediate Code Generation& Code Generation	**09**			
9.	**Intermediate Code Generation** – Introduction, Implementation of Three Address Code, Types of Three Address Statements	3	C,D,I	2	2,5
10.	**Code Generation –Introduction,** Issues in Code Generation - Basic Blocks and Flow Graphs, Dynamic Code Generation and Generating Code form DAGS	3	C,D	2	2,5
11.	Graph-coloring based register allocation, Code Selection	2	C,D	2	2,5
12.	Peephole optimizations	1	C,D	2	1,2
	UNIT IV: Code Optimization	**09**			
13.	Local Optimization and Global Optimization	4	C,D	3,4	2,3
14.	Runtime Systems, Garbage collection	3	C	3,4	1,2
15.	Activation trees and records, Organization of an activation record.	2	C	3,4	1,2,5

	UNIT V: Assembly & Error Handling	**09**			
16.	Internal and External Address Resolution	4	C	4,5	2,3,4
17.	Error Handling	2	C	4,5	2,4
18.	Compiler Error Recovery, Run-Time Errors	3	C	4,5	2,3
	Total contact hours	45			

LEARNING RESOURCES	
Sl. No.	**TEXT BOOKS**
1.	Compilers: Principles, Techniques, and Tools by Alfred V.Aho,MonicaS.Lam,RaviSethi,JeffreyD.Ullman,Pearson Publishers,2008
2.	William M. Waite and Gerhard Goos. Compiler Construction. Springer-Verlag, New York, 2013.
	REFERENCE BOOKS/OTHER READING MATERIAL
3.	Allen I. Holub "Compiler Design in C", Prentice Hall of India, 2003.
4.	J.P. Bennet, "Introduction to Compiler Techniques", Second Edition, Tata McGraw-Hill, 2003.
5.	Kenneth C. Louden, "Compiler Construction: Principles and Practice", Thompson Learning, 2003
6.	HenkAlblas and Albert Nymeyer, "Practice and Principles of Compiler Building with C", PHI, 2001.

Course nature				**Theory**			
Assessment Method (Weightage 100%)							
In-semester	**Assessment tool**	Cycle test I	Cycle test II	Cycle Test III	Surprise Test	Quiz	**Total**
	Weightage	**10%**	**15%**	**15%**	**5%**	**5%**	**50%**
					End semester examination Weightage :		**50%**

15SE327E	Distributed Operating Systems	L	T	P	C
		3	0	0	3
Co-requisite:	NIL				
Prerequisite:	NIL				
Data Book / Codes/Standards	NIL				
Course Category	P PROFESSIONAL ELECTIVE				
Course designed by	Department of Software Engineering				
Approval	32nd Academic Council Meeting, 23rd July, 2016				

PURPOSE	Provides essential concepts of the principles of distributed systems and the functionalities of distributed operating system.

INSTRUCTIONAL OBJECTIVES		**STUDENT OUTCOMES**						
At the end of the course, student will be able to								
1.	Recognize the essential concepts of distributed system.	a	e					
2.	Comprehend about the communication that takes place in Distributed systems	a	b	e				
3.	Realize the necessity of synchronization, consistency and Fault tolerance in a Distributed System.	a	e					
4.	Value the Process management ,File systems, Shared memory	a	b	e				
5.	Acquire apparent scheme regarding distributed object oriented based systems	a	e					

Session	Description of Topic	Contact hours	C-D-I-O	IOs	Reference
	UNIT I:FUNDAMENTALS OF DISTRIBUTED SYSTEMS	**8**			
1.	Introduction to distributed systems, Goals of Distributed Systems	2	C	1	1,2
2.	Hardware Concepts-Bus-based, switched multiprocessors, Bus-based ,Switched micro computers	2	C	1	1
3.	Software Concepts-Network Operating Systems, True Distributed System and Time sharing Multiprocessor Operating System	2	C	1	1,2
4.	Design issues of distributed systems	2	C	1	1,2
	UNIT II:COMMUNICATION IN DISTRIBUTED SYSTEMS	**9**			
5.	Fundamentals of Communication systems, Layered Protocols	2	C	1,2	1
6.	ATM networks	1	C	1,2	1,2
7.	Client Server model - Blocking Primitives and Non-Blocking Primitives,	1	C	1,2	1
8.	Buffered Primitives and Un buffered Primitives	1	C	1,2	1,2
9.	Reliable and Un reliable primitives	1	C	1,2	1,2
10.	Message Passing	2	C,I	1,2	2
11.	Remote Procedure Call	1	C,I	1,2	1
	UNIT III: SYNCHRONIZATION IN DISTRIBUTED SYSTEMS	**9**			
12.	Clock Synchronization - Logical, Physical clocks, clock synchronization algorithms	2	C,I	1,3	1,4,5
13.	Mutual Exclusion – Centralized,Distributed,Token ring algorithms, comparison of three algorithms	2	C,I	1	1
14.	Election Algorithms – The Bully algorithm, ring algorithm	2	C,I	1	1,4
15.	Atomic transactions	1	C,I	1	1
16.	Deadlock prevention and detection in distributed systems	2	C,I	1	1

	UNIT IV: PROCESSES ,FAULT TOLERANCE,DISTRIBUTED SHARED MEMORY	**12**			
17	Threads, System models	1	C	1	1,2
18	Processor Allocation-Allocation models, Design issues for processor allocation algorithm, Processor allocation algorithms	4	C,D,I	1,4	1,3
19	Scheduling in Distributed systems	2	C,D,I	1,4	1,2,3
20	Fault tolerance- Component faults, system failures, synchronous versus asynchronous systems, fault tolerance using active replication and primary backup	3	C,D	1,3	1,2,4
21	Consistency protocols, Page based distributed shared memory	2	C,D	1,3	1,2
	UNIT V: DISTRIBUTED OBJECT BASED SYSTEMS, CASE STUDY	7			
22	Distributed object based systems- DOO Architecture, DOO Process, DOO Communication, Synchronization in Object Based Systems	4	C	1,5	1,2
23	Distributed File Systems	1	C	1,4	1,2
24	Case Study: Ameoba,MachOS,chorus,V-System	2	C	3,4,5	1,2
	Total contact hours	45			

LEARNING RESOURCES	
Sl. No.	**TEXT BOOKS**
1.	Andrew S. Tanenbaum, "*Distributed Operating Systems*", PearsonEducation, Reprint , 2011
2.	Pradeep K. Sinha, "*Distributed Operating Systems Concepts and Design*",PHI, 2012.
	REFERENCE BOOKS/OTHER READING MATERIAL
3.	MukeshSinghal, Niranjan G Shivratri "Advanced Concepts in Operating Systems", McGraw Hill International 2011.
4.	http://www.seas.gwu.edu/~jstanton/courses/cs251/
5.	http://cse.yeditepe.edu.tr/~sbaydere/courses_new/cse532/

Course nature				**Theory**			
Assessment Method (Weightage 100%)							
In-semester	**Assessment tool**	Cycle test I	Cycle test II	Cycle Test III	Surprise Test	Quiz	**Total**
	Weightage	**10%**	**15%**	**15%**	**5%**	**5%**	**50%**
End semester examination Weightage :							**50%**

15SE328E	Programming In PHP	L	T	P	C
		2	2	0	3
Co-requisite:	NIL				
Prerequisite:	NIL				
Data Book / Codes/Standards	NIL				
Course Category	P PROFESSIONAL ELECTIVE				
Course designed by	Department of Software Engineering				
Approval	32nd Academic Council Meeting, 23rd July, 2016				

PURPOSE	PHP and MySQL are the emerging open source technology to construct web application. PHP provided with huge set of standard library and function which aids us to build and deploy application in repaid fashion. Delegate students a good practical knowledge of how to write successful HTML/PHP code utilizing a MySQL database

INSTRUCTIONAL OBJECTIVES		**STUDENT OUTCOMES**						
At the end of the course, student will be able to								
1.	Acquaint themselves with the fundamental concepts and programming environment of PHP.	a	e					
2.	Design classes and efficiently use PHP functions	b	c					
3.	Implement object oriented concepts like inheritance, reusability, and encapsulation.	c						
4.	Apply custom exceptions and employ concurrency.	a						
5.	Understand and Design the Database using MySQL.	a	k					

Session	Description of Topic	Contact hours	C-D-I-O	IOs	Reference
	UNIT I: PHP FUNDAMENTALS	**07**			
1	PHP – Exploring the PHP Environment – HTML Embedding	1	C	1	1,5
2	Comments - Variables	1	C	1	1,5
3	Data types – Operators – PHP String functions	1	C	1	1,5
4	Controls Structures	1	C	1	1,5
5	Arrays – Types – Multi dimension array – Array functions	2	C,I	1	1,5
6	Functions in PHP	1	C,I	1	1,5
	UNIT II: PHP OO LANGUAGE	**06**			
7	Introduction – Object, Class, new Keyword, Constructor, Destructor	1	C,D	1,4	1
8	Accessing Methods and Properties Using the$this Variable	1	D	4	1
9	Class Constants, Cloning Objects, polymorphism	1	C	4	1
10	parent :: and self :: , instanceof Operator, Abstract method and Classes	1	C,D	4	1
11	Interfaces and Inheritance of Interfaces. Final methods, Overloading	1	D	3	1
12	Exception handling	1	C,D	4	1
	UNIT III: WEB PAGES WITH PHP	**06**			
13	Embedding into HTML, User Input, Safe Handling user Input	1	C	1,3	1
14	PHP Form, form processing	1	C	1	1
15	Working with Form Data, GET, POST, REQUEST	1	C	4	1
16	Reading Data in web Pages	1	C,I	4	1
17	Performing Data validation, required data, number , text	1	C,I	4	1
18	Cookies and Session in PHP`	1	C,D	4	1
	UNIT IV: WORKING WITH DATABASE	**06**			
19	MySQL, Creating Database and Table, CURD	2	C	4,5	1,2,3

20	JOIN, Aggregate Queries	1	C,I	2,4	1,2,3
21	Connecting to MySQL with PHP, Accessing and Updating Database with PHP	1	C,I	4	1,2,3
22	SQL injections	1	C	4	1,4
23	Prepared Statements	1	C	4	1
	UNIT V: ADVANCED CONCEPTS	**05**			
24	File Handling -Create, Open, read, write to files	1	C,I	4	1
25	Working with FTP in PHP	1	C,I	4	1
26	PHP mail functions, Advanced mail functions	1	C	2,3	1,4
27	Building and Formatting dates and times	1	C	2,5	1
28	PHP filters	1	C	2,5	1
Total contact hours		30			

Note: Students are given individual application, for that they have to do the case study specified in each unit.

DETAILS OF TUTORIALS					
Session	**Description of Topic**	**Contact hours**	**C-D-I-O**	**IO**	**Reference**
1	Installing and Configuring XAMPP or WAMPP Web Server	2	O	1	1
2	Installing and Configuring MySQL Server	2	O	1	1
3	Case Study: For the given application, the students has to implement the classes and methods.	4	D,I	2,3	1
4	Case Study: Create and includes the necessary exception handling statements inside the applications	2	D,I	4	1
5	Case Study: Create the webpages for the application	4	D,I	2	1
6	Case Study: implement validation for the web pages using PHP Scripts	2	D,I	1	1
7	Case Study: Implement the Session and cookies to manage the user sessions	2	D,I	1	1
8	Case Study: Create the necessary database and tables for the given application	2	D,I	5	1
9	Case Study: Write the code to interface the database with the web application	2	D,I	5	1
10	Case Study: Implement the file handling to enable user to upload and download files	2	D,I	1	1
11	Case Study: Implement the graphing function using PHP Library	2	D,I	1	1
12	Case Study: Implement the mailing to enable the user to communicate	2	D,I	1	1
13	Case Study: implement the PHP filters in the application and deploy the same in XAMPP	2	D,I	1	1
Total Contact Hours		30			

LEARNING RESOURCES	
Sl. No.	**TEXT BOOKS**
1.	Steven Holzner, “PHP: The Complete Reference”, Tata McGraw Hill Education, 1st Edition,2007.
	REFERENCE BOOKS/OTHER READING MATERIAL
2.	Luke Welling, Laura Thomson, “PHP and MySQL Web Development”, Pearson, 4th Edition, 2009.
3.	Larry Ullman, “PHP and MySQL for Dynamic Web Sites”, Prentice Hall, 4th Edition, 2016.
4.	George Schlossnagle,“Advanced PHP Programming”, First Edition, Sams Publishing, 2004.
5.	http://www.w3schools.com/php/

Course nature	**Theory**

Assessment Method – Theory Component (Weightage 50%)					
In-semester	Assessment tool	Cycle Test 2	Cycle Test 2	Mini Project	Total
	Weightage	10%	15%	25%	50%
				End semester examination Weightage :	50%

15SE329E	Visual Programming	L	T	P	C
		2	2	0	3
Co-requisite:	NIL				
Prerequisite:	NIL				
Data Book / Codes/Standards	NIL				
Course Category	P PROFESSIONAL ELECTIVE				
Course designed by	Department of Software Engineering				
Approval	32nd Academic Council Meeting, 23rd July 2016				

PURPOSE	To acquire knowledge in windows and visual programming concepts							
INSTRUCTIONAL OBJECTIVES		**STUDENT OUTCOMES**						
At the end of the course, student will be able to								
1.	Understand the basics of C# and .NET framework	a						
2.	Develop applications using object-oriented aspects of C#	a	b					
3.	Design Windows applications	a	b	e				
4.	Create Database applications using ActiveX Data Objects	a	b	e				
5.	Develop Web based applications	a	b	e				

Session	Description of Topic	Contact hours	C-D-I-O	IOs	Reference
	UNIT I: INTRODUCTION TO C#	**7**			
1.	Introducing C# - Understanding .NET Framework	1	C	1	1
2.	Overview of C# - Literals – Variables Data Types – Operators – Constants - Expressions	1	C	1	1
3.	Program Control Statements: Branching, Looping	1	C,I	1	1
4.	Casting - Methods	1	C,I	1	1
5.	Arrays: Array Class, Array List	1	C,I	1	1
6.	Strings: String, String Builder	1	C,I	1	1
7.	Structures - Enumerations	1	C,I	1	1
	UNIT II: OBJECT ORIENTED ASPECTSOF C#	**6**			
8.	Class – Objects, Constructors – Types of Constructors	1	C	2	1,2
9.	Inheritance and its types	1	C,I	2	1,2
10.	Indexers and Properties	1	C	2	1,2
11.	Polymorphism – Operator and Method Overloading	1	C,I	2	1,2
12.	Interfaces, Abstract Class	1	C,I	2	1,2
13.	Event Handling, Errors and Exception Handling	1	C,I	2	1,2
	UNIT III: APPLICATION DEVELOPMENT ON .NET	**6**			
14.	Building Windows Application	1	C,I	3	2,3
15.	Creating Window Forms with Events and Controls	1	C,D,I	3	2,3
16.	Menu and Toolbar	1	C,I	3	2,3
17.	Delegates - Inheriting Window Forms	1	C,I	3	2,3
18.	SDI and MDI Application	1	C	3	2,3
19.	Dialog Box: Modal and Modeless	1	C,I	3	2,3
	UNIT IV: DATABASE APPLICATION ON .NET	**5**			
20.	Accessing data with ADO.NET: DataSet, Typed Dataset, Data Adapter	1	C,D,I	4	2,3
21.	Updating Database using Stored Procedures	1	C,I	4	2,3
22.	SQL Server with ADO.NET	1	C,D,I	4	2,3
23.	Handling Exceptions – Validating Controls	1	C,I	3	2,3
24.	Windows Application Configuration	1	C	3	2,3
	UNIT V: WEB BASED APPLICATION DEVELOPMENT ON .NET	**6**			
25.	Programming Web Application with Web Forms	1	C	5	2,5
26.	Introduction to ASP.NET, Working with XML and .NET	1	C,I	5	2,5
27.	Creating Virtual Directory and Web Application	1	C,D,I	5	2,5
28.	Session Management	1	C,D,I	5	2,5

29.	Web Services – web.config, Passing Datasets and Returning Datasets from Web Services	1	C,I	5	2,5
30.	Transaction Handling, Exception Handling – Returning Exceptions from SQL Server	1	C,I	5	2,5
	Total contact hours	30			

LEARNING RESOURCES	
Sl. No.	**TEXT BOOKS**
1.	Herbert Schildt, "The Complete Reference: C# 4.0", Tata McGraw Hill, 2012.
2.	Christian Nagel et al. "Professional C# 2012 with .NET 4.5", Wiley India, 2012.
	REFERENCE BOOKS/OTHER READING MATERIAL
3.	Andrew Troelsen , "Pro C# 2010 and the .NET 4 Platform, Fifth edition, APress, 2010.
4.	Ian Griffiths, Matthew Adams, Jesse Liberty, "Programming C# 4.0", Sixth Edition, O'Reilly, 2010.
5.	Matthew MacDonald, "Beginning ASP.NET 4.5 in C#", APress, 2012
6.	Stephen C. Perry, "Core C# and .NET", Prentice Hall, 2005

S. No.	**Description of Experiments**	**Contact hours**	**C-D-I-O**	**IOs**	**Reference**
1.	Simple programs for implementing Control Structures	2	C,I	1	1-6
2.	Programs to implement Arrays, Structures and Strings	4	C,I	1	1-6
3.	Programs to implement console applications using • Properties • Inheritance • Interface • Polymorphism • Exception Handling	4	C,D,I	2	1-6
4.	Programs to implement Event and Delegates	2	C,I	3	1-6
5.	Programs to build windows applications using • MDI • Dialogs • Windows controls	4	C,D,I	3	1-6
6.	Program to access database with ADO.NET	2	C,D,I	4	1-6
7.	Program to implement Data binding controls	2	C,D,I	4	1-6
8.	Programs to implement ASP.Net Validation controls and handling exceptions	2	C,I	4	1-6
9.	Programs to implement a) Reading and Writing XML files b) XML Validation C) Database Table to XML and XML to Table Conversion	2	C,I	1,5	1-6
10.	Program to develop a Web service	2	C,D,I	5	1-6
11.	Program to implement Master and content pages	2	C,D,I	3,5	1-6
12.	Design a Web page using ASP.NET	2	C,D,I	3,4,5	1-6
	Total Contact Hours	**30**			

Course nature					**Theory + Practical**
Assessment Method – Theory Component (Weightage 50%)					
In-semester	**Assessment tool**	Cycle test I	Cycle test II	**Tutorial Project**	**Total**
	Weightage	**10%**	**15%**	**25%**	**50%**
End semester examination Weightage :					**50%**

15SE331E	**Agile Software Process**	**L**	**T**	**P**	**C**
		3	**0**	**0**	**3**

Co-requisite:	NIL
Prerequisite:	NIL
Data Book / Codes/Standards	NIL
Course Category	P PROFESSIONAL ELECTIVE
Course designed by	Department of Software Engineering
Approval	32nd Academic Council Meeting, 23rd July, 2016

PURPOSE	This course imparts knowledge to students in the basic concepts of Agile Software Process, methodology and its development							
INSTRUCTIONAL OBJECTIVES		**STUDENT OUTCOMES**						
At the end of the course, student will be able to								
1	Understand the basic concepts of Agile Software Process.	a	j	k				
2	Comprehend various Agile Methodologies	a	d	j	k			
3	Develop Agile Software Process	a	j	k				
4	Apply principles of Agile Testing	a	e	k				

Session	Description of Topic	Contact hours	C-D-I-O	IOs	Reference
	UNIT I: INTRODUCTION	**9**			
1.	Introduction : Software is new product development	1	C	1	1
1.	Iterative and Evolutionary Methods	3	C	1	1
2.	Agile – Agile Development	3	C	1	1
3.	Case Study: Perform a comparative Study between Traditional / Heavy weight Methodologies with Agile Methodology and give the Key features and Limitations with some sample projects.	2	C	1	1,6
	UNIT II: AGILE AND ITS SIGNIFICANCE	**9**			
4.	Agile Story :Evolutionary delivery ,Scrum Demo, Planning game, Sprint back log, adaptive planning	2	C	1,3	1
5.	Agile Motivation – Problems With The Waterfall - Research Evidence	3	C	1,3	1
6.	Scrum : Method Overview ,Life cycle phases and Work product roles and practices	4	C	1,3	1
	UNIT III: AGILE METHODOLOGY	**9**			
7.	Extreme Programming : Method Overview ,Life cycle phases and Work product roles and practices	2	C	1,2	1
8.	Unified process : Method Overview ,Life cycle phases and Work product roles and practices	2	C	1,2	1
9.	EVO : Method Overview ,Life cycle phases and Work product roles and practices	2	C	1,2,3	1
10.	Case Study: Student group must collaborate and report together along with assigned batch members. Collect the requirements from the client and adopt the suitable agile practice method for your project	3	C,D,I,O	1,2,3	1
	UNIT IV: AGILE PRACTICES	**9**			
11.	Agile Project management	2	C	1	1,2
12.	Agile Environment	2	C	1	1,2
13.	Agile Requirements	2	C	1	1,2
14.	Case Study – Practices : At the end of each sprint, The team Should perform the following • Report weekly deliveries of completed stories about the application to your customer. • Consolidated Documents from Daily scrum , Demonstrations and Reviews • Perform reporting of automated and acceptance tests • Test driven development • Continuous Integration	3	D,I,O	2,3,4	1,5

	UNIT V: AGILITY AND QUALITY ASSURANCE	**9**			
15.	Agile product development – Agile Metrics – Feature Driven Development (FDD)	4	C	4	3,4
16.	Agile approach to Quality Assurance	2	C	4	3,4
17.	Test Driven Development – Agile approach in Global Software Development	3	C	4	3,4
	Total contact hours	45			

LEARNING RESOURCES	
Sl. No.	**TEXT BOOKS**
1.	Craig Larman, "Agile and Iterative Development – A Manager's Guide", Pearson Education – 2006
2.	Lisa Crispin,Janet Gregory, Mike Cohn,BrainMarick, "Agile Testing: A practical guide for Testers and Agile Teams", Addison-wesley publication, 2009.
	REFERENCE BOOKS/OTHER READING MATERIAL
3.	Chetankumar Patel, Muthu Ramachandran, Story Card Maturity Model (SMM): A Process Improvement Framework for Agile Requirements Engineering Practices, Journal of Software, Academy Publishers, Vol 4, No 5 (2009), 422-435, Jul 2009.
4.	Hazza& Dubinsky, Agile Software Engineering, Series: Undergraduate Topics in Computer Science, Springer 2009
5.	Kevin C. Desouza, Agile information systems: conceptualization, construction, and management, Butterworth-Heinemann, 2007.
6.	Agile Software Development – Wikipedia.

Course nature				**Theory**			
Assessment Method (Weightage 100%)							
In-semester	**Assessment tool**	Cycle test I	Cycle test II	Cycle Test III	Surprise Test	Quiz	**Total**
	Weightage	**10%**	**15%**	**15%**	**5%**	**5%**	**50%**
End semester examination Weightage :							**50%**

15SE332E	**Xml And Webservices**	**L**	**T**	**P**	**C**
		3	**0**	**0**	**3**

Co-requisite:	NIL
Prerequisite:	NIL
Data Book / Codes/Standards	NIL
Course Category	P PROFESSIONAL ELECTIVE
Course designed by	Department of Software Engineering
Approval	32nd Academic Council Meeting, 23rd July, 2016

PURPOSE	To study and highlight the features of different technologies involved in XML , web services and Semantic Web							
INSTRUCTIONAL OBJECTIVES		**STUDENT OUTCOMES**						
At the end of the course, student will be able to								
1.	Understand the basics of XML Concepts	j	k					
2.	Conceive the concepts of web services	j	k					
3.	Comprehend Web service Description Language and Universal Description and Discover Integration Concepts	a	j	k				
4.	Apply methods for constructing and evaluating Web architectures	a	j	k				
5.	Familiarize basics of Semantic Web	j	k					

Session	Description of Topic	Contact hours	C-D-I-O	IOs	Reference
	UNIT I: XML TECHNOLOGY	**9**			
1.	Introduction -XML in context – Fundamentals of XML	1	C	1	1
2.	DTD –XML Schemas	2	C,D,I	1	1,2
3.	X – Files :Xpath,Xpointer and Xlink	2	C,D,I	1	1
4.	Integrating XML with databases	2	C,D,I	1	1
5.	Case study – Create an XML based application and Integrate with database	2	D,I.O	1	1
	UNIT II: ARCHITECTING WEB SERVICES	**9**			
6.	Business motivations for web services	1	C	1,2	1
7.	Service oriented Architecture (SOA)	3	C	2,4	1
8.	Architecting Web services :Implementation ,Logical and Technological views	5	C	2,4	1
	UNIT III: WEB SERVICES BUILDING BLOCK	**9**			
9.	Simple Object Access Protocol (SOAP)	2	C,D	2,3	1,3
10.	Web service Description Language (WSDL) ,Universal Description and Discovery Integration (UDDI)	3	C,D	2,3	1,3,4
11.	Case study : Create a Web service for a specific application using WSDL ,UDDI and SOAP concepts	4	D,I	2,3	1,2,3
	UNIT IV: XML IN E – BUSINESS	**9**			
12.	B2B – B2C Applications – Different types of B2B interaction	2	C	2	1
13.	Enterprise Integration – eb XML	3	C	2,4	1
14.	Rosetta Net -Applied XML in vertical industry	4	C	2,4	1
	UNIT V: SEMANTIC WEB	**9**			
15.	Semantic Web – Role of Meta data in web content	2	C	2,5	1
16.	Resource Description Framework – RDF schema	4	C	2,5	1
17.	Architecture of semantic web	3	C	2,5	1
	Total contact hours	45			

LEARNING RESOURCES

Sl. No.	**TEXT BOOKS**
1.	Ron Schmelzer et al, "XML and Web Services Unleashed", Pearson Education, 2014.
2.	Frank P.Coyle, "XML, Web Services and the Data Revolution", Pearson Education, 2010
	REFERENCE BOOKS/OTHER READING MATERIAL
3.	Russ Basiura and Mike Batongbacal, "Professional ASP .NET Web Services", Apress, 2009
4.	Henry Bequet and MeerajKunnumpurath, "Beginning Java Web Services", Apress, 2004

Course nature	**Theory**
Assessment Method (Weightage 100%)	

In-semester	**Assessment tool**	Cycle test I	Cycle test II	Cycle Test III	Surprise Test	Quiz	**Total**
	Weightage	**10%**	**15%**	**15%**	**5%**	**5%**	**50%**
End semester examination Weightage :							**50%**

15SE333E	Pervasive Computing	L	T	P	C
		3	0	0	3
Co-requisite:	NIL				
Prerequisite:	NIL				
Data Book / Codes/Standards	NIL				
Course Category	P PROFESSIONAL ELECTIVE				
Course designed by	Department of Software Engineering				
Approval	32nd Academic Council Meeting, 23rd July, 2016				

PURPOSE	This course provides a way to understand the concepts of WIRELESS LAN, WAP, WML, PDA and its issues.							
INSTRUCTIONAL OBJECTIVES		STUDENT OUTCOMES						
At the end of the course, student will be able to								
1	Understand the fundamental elements of pervasive computing.	b						
2	Learn the design process of Pervasive Computing Environments and its solutions	c						
3	Familiarize hardware, software and the aspects involved in pervasive computing	c	i					
4	Comparative study of protocols, languages, models & technologies involved	g						
5	Learn WAP and voice technology.	i	j					

Session	Description of Topic	Contact hours	C-D-I-O	IOs	Reference
	UNIT I - INTRODUCTION	10			
1.	Pervasive Computing: Past, Present and Future Pervasive computing,	1	C	1	1,2
2.	Pervasive Computing Market, m-Business	2	C	1	1,2
3.	Application examples: Retail, Airline check-in and booking, Healthcare	1	C,I	1,2	1,2
4.	Tracking, Car information system, Sales Force Automation, E-mail access via WAP and voice	1	C,I	1,2	1
5.	Device Technology: Hardware, Human machine interface	1	C	3	1
6.	Bio metrics, Operating systems	2	C	3	1
7.	Java for pervasive devices	2	C	4	1
	UNIT II DEVICE CONNECTIVITY & WEBAPPLICATION CONCEPTS	9			
8.	Device connectivity : Protocols: wireless, mobile phone technologies, mobile internet protocol	2	C	3	1,3
9.	Synchronization and replication protocol, distributed services and message and transaction protocols	2	C	4	1,3
10.	Security	2	C	3	1,3
11.	Device Management	1	C	3	1
12.	Web Application Concepts: WWW Architecture and Protocols	1	C	4	1
13.	Transcoding , Client Authentication via Internet	1	C	4	1
	UNIT IIIWAP & VOICE TECHNOLOGY	8			
14.	WAP and Beyond: Introduction, Components of the WAP architecture	1	C	5	1
15.	WAP infrastructure, WAP security issues	1	C,O	5	1
16.	Wireless Markup Language , WAP push	2	C,D	4,5	1
17.	Products,i-Mode	1	C	5	1
18.	VoiceTechnology: Basics of Speech recognition, Voice Standards	2	C,I	5	1
19.	Speech Applications, Speech and Pervasive Computing, security	1	C,I	5	1
	UNIT IV PDA & PERVASIVE WEB APPLICATION	9			

	ARCHITECTURE				
20.	Device Categories, PDA operation Systems	2	C	3	1
21.	Device Characteristics, SoftwareComponents	1	C	3	1
22.	Standards, Mobile Applications, PDA Browsers	2	C	3	1
23.	Pervasive WebApplication architecture: Background, Scalability and availability, Development of Pervasive Computing webapplications	2	C	2,3	1
24.	Pervasive application architecture.	2	C	2	1
	UNIT V –APPLICATION IN PERVASIVE COMPUTING	9			
25.	User Interface Issues in Pervasive Computing, Architecture	3	C	2	1
26.	Smart Card- based Authentication via internet and ordering goods	2	C,I	2	1
27.	Access from WAP	2	C,I	2	1
28.	Access from personal digital assistants	2	C,I	2	1
29.	Access via voice	2	C,I	2	1
	Total contact hours	45			

LEARNING RESOURCES	
Sl. No.	**TEXT BOOKS**
1.	JochenBurkhardt, Horst Henn, Stefan Hepper, Thomas Schaech & Klaus Rindtorff, "Pervasive Computing, Technology and Architecture of Mobile Internet Applications", Pearson Education, 2012. ISBN-13**:** 978-0201722154
	REFERENCE BOOKS/OTHER READING MATERIAL
2.	UweHansmann, L. Merk, M. Nicklous, T. Stober, U. Hansmann, "PervasiveComputing (Springer Professional Computing) ", 2003, Springer Verlag,ISBN:3540002189.
3.	Frank Adelstein, Sandeep KS Gupta, Golden Richard III, Loren Schwiebert, "Fundamentals of Mobile and Pervasive Computing", McGraw Hill edition, 2006. ISBN-13**:** 978-0071412377

Course nature				**Theory**			
Assessment Method (Weightage 100%)							
In-semester	Assessment tool	Cycle test I	Cycle test II	Cycle Test III	Surprise Test	Quiz	Total
	Weightage	10%	15%	15%	5%	5%	50%
End semester examination Weightage :							50%

15SE334E	Advanced Java Programming	L	T	P	C
		2	2	0	3
Co-requisite:	NIL				
Prerequisite:	15SE205J				
Data Book / Codes/Standards	NIL				
Course Category	P PROFESSIONAL ELECTIVE				
Course designed by	Department of Software Engineering				
Approval	32nd Academic Council Meeting, 23rd July, 2016				

PURPOSE	Having a hands on core java programming concepts, this course reviews advanced concepts in programming that motivate the students to build innovative applications. This course explores the skills required to develop J2EE enterprise applications using the Java programming language.

INSTRUCTIONAL OBJECTIVES		**STUDENT OUTCOMES**						
At the end of the course, student will be able to								
1	Design interactive applications with GUI Components such as Java FX.	d						
2	Make effective use of Java networking API to communicate between processes using network sockets	c						
3	Develop database applications using JDBC	c						
4	Designing Java Sever Pages and Java Servlet	i						
5	Understand Java Reflection API and XML DOM Parser related to Industry trends.	c						

Session	Description of Topic (Theory)	Contact hours	C-D-I-O	IOs	Reference
	UNIT I: JAVA GUI PROGRAMMING USING JAVA FX	**6**			
1	Basics of Java FX	1	C	1	1-5
2	Java FX and Containers	2	C	1	1-5
3	Frames, Layout Manager	1	C	1	1-5
4	Menus and Toolbars	1	C	1	1-5
5	Event Handling	1	C	1	1-5
6	Review of Unit I	1	C	1	1-5
	UNIT II: JAVA DATABASE CONCEPTS(JDBC 4)	**6**			
7	Database Architecture : Components of JDBC – Two Tier/ Three Tier Architecture Processing SQL Statements	1	D	3	1-5
8	Establish Connection : [Using Driver Manager class, Connection URLs], Types – Concurrency – Read column values from rows – Updating rows in a result set	1	D	3	1-5
9	Create a statement – Execute Query – Process Result Set, Reading from and modifying values in a Result Set	1	D	3	1-5
10	Handling SQL Exceptions : Contents of SQLException object , Retrieving warnings using SQLWarning object	1	D	3	1-5
11	Statements Creating Prepared Statement object – Assign values for Prepared Statement parameters	2	I	3	1-5
	UNIT III: JAVA NETWORKING	**6**			
12	URL: Creating and Parsing URL – URLConnection : Connecting to a URL	1	D	2	1-5
13	Reading from and Writing to aURLConnection.	1	C	2	1-5
14	Socket: Server Socket/Socket class	1	C	2	1-5
15	InetAddress. Datagram's: Writing a datagram client and server	1	C	2	1-5
16	Datagram Socket, Datagram Packet – Broadcasting tomultiple recipients	1	D	2	1-5
17	Multicast Socket-SSL and HTTPS in Java,RMI	1	D	2	1-5

	UNIT IV: JAVA SERVER SIDE TECHNOLOGIES	**6**			
18	Overview of JSP2.2 and Servlet 3.1	1	C	4	1-5
19	Creating dynamic WebPages using JSP and Servlet- Standard Tag Library	1	I	4	1-5
20	Database Access- XML Data	1	I	4	1-5
21	Java Beans - Custom Tags - Expression Language – Annotations	1	C	4	1-5
22	Filters-Event handling-Exception Handling - Asynchronous processing	1	I	4	1-5
23	Debugging - Security – Internationalization	1	C	4	1-5
	UNIT V: REFLEXTION API & JAVA XML	**6**			
24	Introduction to Java Reflextions API	1	I	5	1-5
25	Introspection	1	I	5	1-5
26	Dynamic Proxies	1	I	5	1-5
27	Dynamic class loading and reloading	1	I	5	1-5
28	Java XML: XML Processing	1	I	5	1-5
29	SAX and DOM Parser	1	I	5	1-5
	Total Hours	30			

Sl. No.	**Description of Tutorials**	**Contact hours**	**C-D-I-O**	**IOs**	**Reference**
1.	Programs on Java FX basic components with containers and frames	3	I	1	1-5
2.	Programs on Java FX menu bars and tool bars	3	D	2	1-5
3.	Programs on TCP and UDP Socket	3	D	2	1-5
4.	Programs on Multicast Socket Applications	3	I	2	1-5
5.	Programs on Java RMI Concepts	3	I	2	1-5
6.	Programs on JDBC data base connectivity and normal execution of queries	3	D	3	1-5
7.	Programs on Result Set iteration and Prepared Statements	3	I	4	1-5
8.	Programs on designing pages using JSP	3	D	4	1-5
9.	Programs on MVC Based Web Applications	3	D	5	1-5
10.	Programs on SAX and DOM Parser.	3	I	5	1-5
	Total contact hours	30			

Sl. No.	**LEARNING RESOURCES**
1	Carl Dea, Mark Heckler, GerritGrunwald, José Pereda, Sean Phillips *"JavaFX 8: Introduction by Example" Apress 2nd Edition 2014*
2	Kogent Learning Solutions Inc, "Java Server Programming Java EE 7 (J2EE 1.7), Black Book", dreamtechpress 2015
3	Elliotte Rusty Harold,"Java Network Programming, 4th Edition Developing Networked Applications "O'Reilly Media, Final Release Date: October 2013
4	Paul Deital,HarveyDeital,"Java How to Program. "10th Edition,2015
5	http://pdf.coreservlets.com/
6	https://docs.oracle.com/javase/tutorial

Course nature				**Theory**	
Assessment Method – Theory Component (Weightage 50%)					
In-semester	**Assessment tool**	**Cycle Test I**	**Cycle Test 2**	**Tutorial (Mini Project)**	**Total**
	Weightage	**10%**	**15%**	**25%**	**50%**
				End semester examination Weightage :	**50%**

15SE421E	**Software Reliability**	**L**	**T**	**P**	**C**

		3	**0**	**0**	**3**
Co-requisite:	NIL				
Prerequisite:	15SE202				
Data Book / Codes/Standards	NIL				
Course Category	P PROFESSIONAL ELECTIVE				
Course designed by	Department of Software Engineering				
Approval	32nd Academic Council Meeting, 23rd July, 2016				

PURPOSE	This course will give an insight at the professional techniques for assessing and modeling the reliability of software systems.							
INSTRUCTIONAL OBJECTIVES		**STUDENT OUTCOMES**						
At the end of the course, student will be able to								
1.	Conceive the concept of software reliability	e						
2.	Understanding of Reliability Estimation Methods.	a	e					
3.	Implement reliability testing procedures.	b	e	k				
4.	Adapt and apply software reliability models.	b	k					
5.	Develop and apply a fault prediction models for large systems.	b						

Session	Description of Topic	Contact hours	C-D-I-O	IOs	Reference
	UNIT I: INTRODUCTION TO SOFTWARE RELAIBILITY	**8**			
1.	The Need for System Software Reliability, Software-related Problems	2	C	1	1
2.	Importance of Reliability Engineering – REAL TIME CASE STUDY	1	C,D,I,O	1	1
3.	System Reliability with Multiple Failure Modes	2	C,D	1	1
4.	Reliability Calculations	1	C,D	1	1
5.	Reliability Measures - MTTF, MTTD,MTBF,MTTR	1	C,D	1	1
6.	Case Study: Future Problems in the Twenty-first Century and impact of software reliability	1	C,D,I,O	1	1
	UNIT II: ESTIMATION METHODS FOR RELIABILITY ANALYSIS	**9**			
7.	Point Estimation	1	C	2	1
8.	Maximum Likelihood Estimation Method	1	C	2	1
9.	Confidence Intervals of Estimates	1	C,D,I	2	1
10.	Statistical Change-point Estimation Methods - Goodness of Fit Techniques	2	C,D,I,O	2	1
11.	Chi-squared Test & Least Squared Estimation	2	C,D,I,O	2	1
12.	Interval Estimation using Sequential Sampling	1	C,D,I,O	2	1
13.	Implementation of Bayesian Methods using a real time scenario	1	C,D,I,O	2	1
	UNIT III: RELAIBILITY IN SOFTWARE SYSTEMS	**9**			
14.	Software *vs*Hardware Reliability, Software Reliability and Testing Concepts of Reliability	1	C	3	1
15.	Introduction to Testing Coverage Models, Testing Coverage and Imperfect Debugging	2	C,D	1,2	1
16.	Fault Removal Efficiency Model - Model Implementations	1	C,D,I,O	1,2	1
17.	Imperfect Debugging Model with Multiple Failure Types	1	C	1,2	1
18.	A Constant Fault Detection Rate	1	C,D,I,O	1,2	1
19.	Fault Detection Time-dependent Rate	1	C,D	1,2	1
20.	Introduction , Halstead's Software Metric, McCabe's Cyclomatic Complexity Metric	2	C,D	4	1,2
	UNIT IV: SOFTWARE RELIABILITY MODELING	**11**			
21.	Failure Rate Models, Curve Fitting Models	1	C,D	4	1,2
22.	Model Selection	1	C	4	1,2
23.	Reliability Growth Models	2	C,D,I	4	1,2

24.	Markov Structure Models, Time Series Models	2	C,D,I	4	1, 2
25.	Non-homogeneous Poisson Process Models (NHPP),	1	C,D,I	4	1, 2, 4
26.	NHPP S-shaped Models ,NHPP Imperfect Debugging Models	2	C,D,I	4	1, 2, 4
27.	Parameter Estimation, NHPP Exponential Models	2	C,D,I,O	4	1, 2 , 4
	UNIT V: RECOVERY SYSTEM RELIABILITY MODELING	**8**			
28.	Mean Time Between Failures for NHPP	2	C,D	1,2	1,3
29.	Enhanced Proportional Hazard Jelinski-Moranda (EPJM) Model	2	C,D,I	1,2	1,3
30.	Gain Model with Random Field Environments	1	C,D,I	1,2	1,3
31.	Calender Time Modeling	1	C	1,2	1,3
32.	Basic Fault-tolerant Software Techniques	2	C,D	1,2	1,3
	Total contact hours	45			

LEARNING RESOURCES	
Sl. No.	**TEXT BOOKS**
1.	Hoang Pham, " System Software Reliability", Springer Series, 2013.
	REFERENCE BOOKS/OTHER READING MATERIAL
2.	M. Xie, "Software Reliability Modelling" , Tata McGraw Hill, 2005.
3.	*Handbook of Software Reliability Engineering, Michael Lyu (ed.), 1996, IEEE Computer Society Press, ISBN: 0-07-039400-8*
4.	*(*A Detailed Study of NHPP Software Reliability Models) http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.655.7971&rep=rep1&type=pdf

Course nature				**Theory**			
Assessment Method (Weightage 100%)							
In-semester	**Assessment tool**	Cycle test I	Cycle test II	Cycle Test III	Surprise Test	Quiz	**Total**
	Weightage	**10%**	**15%**	**15%**	**5%**	**5%**	**50%**
End semester examination Weightage :							**50%**

15SE422E	**Software Quality Assurance**	**L**	**T**	**P**	**C**

		3	**0**	**0**	**3**
Co-requisite:	NIL				
Prerequisite:	15SE202				
Data Book / Codes/Standards	NIL				
Course Category	P PROFESSIONAL ELECTIVE				
Course designed by	Department of Software Engineering				
Approval	32nd Academic Council Meeting, 23rd July, 2016				

PURPOSE	To Acquire Knowledge about importance of software quality, Components of software quality assurance system and quality standards							
INSTRUCTIONAL OBJECTIVES		**STUDENT OUTCOMES**						
At the end of the course, student will be able to								
1.	Familiarize with the components to ensure quality in different process levels	a	b					
2.	Apply different quality components in different level of software development	a	e					
3.	Equip themselves familiar with standards	h	i					
4.	Operate different quality tools	k						

Session	Description of Topic	Contact hours	C-D-I-O	IOs	Reference
	UNIT I: INTRODUCTION	8			
1.	Software Quality Concepts and Culture: Importance of Quality, Uniqueness of software quality Assurance, The main Characteristics of SQA environment, Software Quality concepts and definitions, causes of software Errors Differences between software and other industrial products	2	C	1	1,3
2.	Society's concern for quality: Popular view, Professional view	1	C	1	4
3.	Costs and impacts of bad quality- Cost of Quality model: Classic and Extended Models	2	C	1	1
4.	Software Quality Factors - McCalls Model, Alternative models of software quality factor:	3	C,D	1	1,3
a	**Unit II : PROCESS ASSURANCE**	11			
5.	Nature of Process Assurance -Software Quality Assurance System Architecture	2	C	1	1,3
6.	Contract Review process and its stages, Contract Review objectives, Implementation of Contract Review, Contract review for internal projects	1	C, D,O	1,2	1,2
7.	Development plan, Quality Plan	2	C, D,O	1,2	1,2
8.	Process Assurance Techniques	1	C	1,2	1,2
9.	Reviews	1	C,D	1,2	1,2
10.	Software Testing –Strategies, process and Implementation	3	D,I	1,2	1,2,5
11.	Distinctions between Assurance Verification and Validation	1	C	1,2	1
a	**UNIT III: PRODUCT ASSURANCE**	9			
12.	Assuring the Quality of Software Maintenance	1	C,D	1,2	1,2,5
13.	Assuring The Quality of External Participants' Parts	1	C	1,2	1
14.	Case Tools and their Affect on Software Quality	1	C	1,4	1
15.	Procedures and Work Instructions	1	C,I	1,2	1
16.	Supporting Quality Devices	1	C,O	1,2	1,4
17.	Configuration Management	2	C,I	1,2	1
18.	Preventive and Corrective Actions	1	C,I	1,2	1

19.	Documentation and Quality Records Control	1	C,I	1,2	1
a.	**UNIT IV: HUMAN TASKS AND METRICS FOR QUALITY**	9			
20.	Management and its Role in Quality Assurance	1	C	1	1
21.	Staff Training Instructing and Certification	`1	C,O	1,2	1
22.	The Software Quality Assurance Unit	1	C	1	1
23.	SQA Trustees -SQA Committees	1	C	1	1
24.	Quality product metrics and Measurements	4	C	1,4	1,2,4
25.	Practicing Quality Metrics with Case Study	1	O	1,4	1,2,4
a.	**UNIT V: QUALITY PRODUCT MODELS**	**8**			
26.	SQA and ISO Standards	1	C	3	1,2,6
27.	SQA and IEEE Standards	1	C	3	1,2,6
28.	SQA and CMM	1	C	3	1,2,4,6
29.	Boehm's Quality Model	1	C	3	1,2,4,6
30.	Other models: FURBS, IBM, Malcolm Baldrige Assessment	3	C	3	1,2,4,6
	Case study to practice Quality factors	1	C,O	2	1
	Total contact hours	45			

LEARNING RESOURCES	
Sl. No.	**TEXT BOOKS**
1.	Daniel Galin, "Software Quality Assurance from theory to implementation", Pearson, 2009.
2.	G.GordonSchulmeyer, "Hand book of Software Quality Assurance", 4th edition, ARTECH HOUSE INC, 2008
	REFERENCE BOOKS/OTHER READING MATERIAL
3.	Allen Gilles, "Software quality: Theory and management" - International Thomson - Computer press, 2011
4.	Stephen H.Kan - "Metrics and models in software quality Engineering" - Addison – Wesley, 2011.
5.	Roger S Pressman, "Software Engineering – A Practitioner's Approach", 7th edition, McGraw Hill Education, 2014.
6.	http://www.bth.se/com/besq.nsf/(WebFiles)/CF1C3230DB425EDCC125706900317C44/$FILE/chapter_1.pdf

Course nature				**Theory**			
Assessment Method (Weightage 100%)							
In-semester	**Assessment tool**	Cycle test I	Cycle test II	Cycle Test III	Surprise Test	Quiz	**Total**
	Weightage	**10%**	**15%**	**15%**	**5%**	**5%**	**50%**
End semester examination Weightage :							**50%**

15SE423E	Software Configuration Management	L	T	P	C
		3	0	0	3
Co-requisite:	NIL				
Prerequisite:	15SE202				
Data Book / Codes/Standards	NIL				
Course Category	P PROFESSIONAL ELECTIVE				
Course designed by	Department of Software Engineering				
Approval	32nd Academic Council Meeting, 23rd July, 2016				

PURPOSE	Software Configuration Management (SCM) is required to control evolving and changing software systems. This course will introduces the basic concepts of software configuration management, the importance of SCM in software development, the different SCM phases and activities, release management, CASE tools used in SCM industry.

INSTRUCTIONAL OBJECTIVES		**STUDENT OUTCOMES**						
At the end of the course, student will be able to								
1.	Understand the different roles of people involved in SCM	a						
2.	Understand how to apply Identify Configuration Change in real cases.	a	b					
3.	Verify and Track full application	b						
4.	Utilize CASE tools to control and document the change.	b	k					
5.	Be aware of SCM models	a						

Session	Description of Topic	Contact hours	C-D-I-O	IOs	Reference
	UNIT I: INTRODUCTION	**9**			
1.	SCM: Concepts and definitions	1	C	1	1,3
2.	Need and importance of SCM	1	C	1	1,3
3.	Benefits of SCM	1	C	1	1,3
4.	Plan preparation with case study	1	D	1	1,3,4
5.	Infrastructure setup	1	D	1	1
6.	Team Training	1	C	1	1
7.	System Operation	1	C	1	1
8.	System Maintenance	1	C	1	1
9.	System Retirement	1	C	1	1
	UNIT II: CONFIGURATION IDENTIFICATION &CONTROL	**9**			
10.	Configuration Item	1	C	2	1
11.	Configuration Identification Process with case study	1	C,D	2	1
12.	Change initiation	1	C,D	2	1
13.	Implementation	1	C,I	2	1
14.	Evaluation-Documented Process, Product Structure, Configuration Documentation, Product Identification, Configuration Baselines, Interface Control, Metrics	2	C,D	2	1,2
15.	change control – Submitting Change Request , Request Analysis, Request Disapproval and Request Approval	2	C,D	2	1,2
16.	Configuration Control Tool	1	C,I	2	1
	UNIT III: CONFIGURATION VERIFICATION & AUDITS	**9**			
17.	Verification and Audit Plan For a case study	1	C,D	3	1,2
18.	Configuration Verification Process	1	C	3	1,2
19.	Pre-Audit, Audit & Post Audit	1	C,D	3	1,2
20.	Problem Reporting and Tracking	1	C,D	3	1,2
21.	Status accounting	1	C	3	1,2
22.	Status accounting Activity Model	1	C	3	1,2
23.	Information gathering & database	1	C	3	1,2
24.	Severity & Prevention	1	C	3	1,2
25.	Change Log, Progress Report , CI Status Report & Transaction Log	1	C	3	1,2

	UNIT IV: VERSION CONTROL & RELEASE MANAGEMENT	**9**			
26.	Version Control Benefit	1	C	4	1,4,5
27.	Version Identification & Storage Management	1	C	4	1,4,5
28.	Change History Recording & Independent Development	1	C	4	1,4,5
29.	System Building Problem	1	C	4	1
30.	System Building Process	1	C,D	4	1
31.	System Library	1	C	4	1
32.	Release Note and Checklist	1	C	4	1
33.	Release Log, User Manual & Release Labe1	1	C	4	1
34.	Deployment Management & Tools	1	C,I	4	1
	UNIT V: SOFTWARE CONFIGURATION MANAGEMENT MODELS	**9**			
35.	SCM standards-IEEE perspective, ISO perspective & SEI perspective	1	C	5	1
36.	Military standards	1	C	5	1
37.	International standards	1	C	5	1
38.	Capability Maturity Model	2	C	5	1
39.	Maturity Model Integration	1	C	5	1
40.	Trillium Model	1	C	5	1
41.	Standard Tool	1	I	5	1
42.	Process Improvement Models Tool	1	I	5	1
	Total contact hours	45			

LEARNING RESOURCES	
Sl. No.	**TEXT BOOKS**
1.	Alexis Leon, "*Software Configuration Management Handbook*", Artech House Publishers, 2004
2.	Anne Hass , Configuration Management Principles and Practice, Addison Wesley, 2002
	REFERENCE BOOKS/OTHER READING MATERIAL
3.	Brad Appleton, Kyle Brown, Stephen P. Berczuk, "*Software Configuration Management Patterns : Effective Teamwork, Practical Integration*", Addison-Wesley, 2002
4.	Sean Kenefick, "*Real World Software Configuration Management*", Apress, 2008
5.	Jessica Keyes, Software Configuration Management, Auerbach Publications, 2004

Course nature				**Theory**			
Assessment Method (Weightage 100%)							
In-semester	**Assessment tool**	Cycle test I	Cycle test II	Cycle Test III	Surprise Test	Quiz	**Total**
	Weightage	**10%**	**15%**	**15%**	**5%**	**5%**	**50%**
End semester examination Weightage :							**50%**

15SE424E	**Analysis Of Software Artifacts**	**L**	**T**	**P**	**C**

		3	**0**	**0**	**3**
Co-requisite:	NIL				
Prerequisite:	15SE202				
Data Book / Codes/Standards	NIL				
Course Category	P PROFESSIONAL ELECTIVE				
Course designed by	Department of Software Engineering				
Approval	32nd Academic Council Meeting, 23rd July, 2016				

PURPOSE	To enhance students software testing and analysis skills.							
INSTRUCTIONAL OBJECTIVES		**STUDENT OUTCOMES**						
At the end of the course, student will be able to								
1.	Understand the concepts of quality models, frame work and testing	e						
2.	Gain knowledge in analysis of software under various dimensions	a	b	c				
3.	Gain knowledge in the area of evaluating an architecture, verification & validation of software.	k						
4.	know what kinds of analyses are available and how to use them;	j	k					
5.	understand their scope and power, when they can be applied, and what conclusions can be drawn from their results;	j						
6.	have a grasp of fundamental notions sufficient to evaluate new kinds of analysis when they are developed;	c	k					
7.	have some experience selecting and writing analyses for a real piece of software, applying them and interpreting the results	c						

Session	**Description of Topic**	**Contact hours**	**C-D-I-O**	**IOs**	**Reference**
	UNIT I: QUALITY MODELS	**9**			
1.	Introduction to software artifacts, needs and uses.	1	C	3, 5	2
2.	Views on quality	1	C	3, 5	2
3.	cost of quality	1	C	3, 5	2
4.	quality models	1	C	3, 5	2,3
5.	Statistics and measurements	1	C	6	6
6.	Analysis of given source code using SQALE and Sonar models.	2	C,D,I,O	1,4,6,7	6
7.	Analysis of given source code using Sonar models.	2	C,D,I,O	1,4,6,7	6
	UNIT II: QUALITY FRAMEWORK and TESTING	**12**			
8.	Quality framework characteristics	1	C,D	2	2
9.	Verification	1	C	3	7,8
10.	Measuring test adequacy	1	C	1,6,7	7,8
11.	Overview of black box testing techniques	1	C,D,I,O	1,6,7	7,8
12.	Decision tables	2	C,D,I,O	1,6,7	7,8
13.	Combinatorial testing	2	C,D,I,O	1,6,7	7,8
14.	Classification tree method	2	C,D,I,O	1,6,7	7,8
15.	White box testing	1	C,D,I,O	1,6,7	7,8
16.	Random and exploratory	1	C	1,6,7	7,8
	UNIT III: SOFTWARE ANALYSIS	**8**			
17.	Introduction to Static analysis	1	C	2,4	15
18.	Static analyzer for finding dynamic programming errors	2	C,D,I,O	2,4	15
19.	Dataflow testing – procedure to apply data flow testing-examples	2	C,D,I,O	1,3	1
20.	Performance analysis and verification	1	C,D,I,O	3,4	18
21.	Security analysis and verification	1	C	3,4	16
22.	Software vulnerabilities and exploitation	1	C	6	17
	UNIT IV: QUASAR METHOD	**7**			
23.	Applying the Design structure matrix to system decomposition and integration problems	3	C,D,I,O	4,5,6,7	4
24.	Achieving Agility through Architecture visibility	2	C	4,5,6,7	4

25.	Recovering and verifying architecture through design structure matrices.	2	C,D	4,5,6,7	4
	UNIT V: QUALITY MANAGEMENT	**9**			
26.	Project quality management :Quality Artifacts and its impact on projects	1	C	3,5	11
27.	Essential Testing : the phases and its importance	2	C,D	3,5	14
28.	Test driven development	2	C	3,5	12, 13
29.	Guidance for software verification and validation plans	2	C	3,5	10
30.	Master Test Planning : the process and procedure	2	C,D,I,O	3,5	9
	Total contact hours	45			

LEARNING RESOURCES	
Sl. No.	**TEXT BOOKS**
1.	KshirasagarNaik and PriyadarshiTripathy, "Software testing and Quality Assurance: theory and practice", John wiley& sons Inc, copyright, 2008.
2.	Daniel Galin, "Software Quality Assurance from Theory to Implementation", Pearson Education Ltd., 2004.
	REFERENCE BOOKS/OTHER READING MATERIAL
3.	"Quality models to engineering quality requirements" published in journal of object technology, chair of Software engineering, Vol.2, No. 5 Sep. – October 2003. Online at http://www.jot.sm.
4.	Tyson R. Browning, A review and new directions, "Applying the design structure matrix to system decomposition and integration problems", IEEE transactions on Engineering management, Vol. 48, No.3, August 2001.
5.	Neerajsangal and frank waldman in the journal of "Defense software engineering Dependency models to manage software Architecutre", Online at www.stsc.hill.af.mil., November 2005.
6.	Stephen Chin, Erik Huddleston, Walter Bodwell and Israel Gat, "The Economics of Technical Debt", published in Cutter IT Journal, Vol.23, No. 10 Mar. – October 2008
7.	Frank Buechner, " is 100% code coverage enough", Hitex Development Tools GmbH, 2008.
8.	James Bach, "Essential Test Design", Fearless Consulting Kb, Unique Publishing, ISBN 13: 978 91 85699039
9.	Joel Montvelisky, "Master Test Plan – the strategic side of testing", Test Process, 2008.
10.	Dolores R Wallace, Roger U Fujii, "software verification and validation plans", IEEE, 1989
11.	"Project Quality Management for Development Organization", pm4dev, 2008 –management for development series
12.	Elisabeth Hendrickson, "Driving Development with Tests: ATDD and TDD", Quality Tree Software, Inc, 2008. Online at http://testobsessed.com/wp-content/uploads/2011/04/atddexample.pdf
13.	Kent Beck, "Test-Driven Development By Example", Three Rivers Institute, 2002. Online at http://www.eecs.yorku.ca/course_archive/2003-04/W/3311/sectionM/case_studies/money/KentBeck_TDD_byexample.pdf
14.	Greg Fournier, "Essential Testing: A Use Case Driven Approach", BookSurge Publishing, 2007.
15.	William R Bush, Jonathan D Pincus and David J Sielaff, " A static Analyzer for finding dynamic programming error", Software-Practice and Experience, 2000.S
16.	Karen Scanrfone, MurugiahSouppaya, Amanda Cody, "Technical Guide to Information Security Testing and Assessment", National Institute of Standard and Technology, 2008.
17.	Peter Mell, Karen Scanrfone, Sasha Rommanosky, " A complete guide to the common Vulnerability Scoring System", Common Vulnerability Scoring System, Version 2.0, 2007.
18.	Shallahmer, "Practical Queuing Theory", Online : link.springer.com/content/pdf/10.1007%2F978-1-4302-0208-0_5.pdf

Course nature				**Theory**			
Assessment Method (Weightage 100%)							
In-semester	**Assessment tool**	Cycle test I	Cycle test II	Cycle Test III	Surprise Test	Quiz	**Total**
	Weightage	**10%**	**15%**	**15%**	**5%**	**5%**	**50%**
					End semester examination Weightage :		**50%**

15SE425E	Software Maintenance And Administration	L	T	P	C
		3	0	0	3
Co-requisite:	Nil				
Prerequisite:	15SE202				
Data Book / Codes/Standards	Nil				
Course Category	P PROFESSIONAL ELECTIVE				
Course designed by	Department of Software Engineering				
Approval	32nd Academic Council Meeting, March, 2016				

PURPOSE	To study how to maintain and administrate the software							
INSTRUCTIONAL OBJECTIVES		**STUDENT OUTCOMES**						
At the end of the course, student will be able to								
1.	Understand the factors that make change of existing systems both technically challenging and risky, and the processes required to control change.	b						
2.	Acquire a knowledge and understanding of the specific problems inherent in the reengineering and evolution of legacy software systems, and be able to apply some of the techniques that can be of use in comprehending and changing them.	b	c					
3.	Identify the specific challenges that inherent in the reengineering and evolution of data-intensive systems that are able to apply some of the techniques that can be of use in comprehending and changing them.	a	b					
4.	Evaluate and understand the specific problems inherent in the reengineering and evolution of package-based software systems, and be able to apply techniques for designing change-resistant systems from pre-packaged code.	e						
5.	Analyze and apply numerous administration tools and technical concepts that relate to software administration	k						

Session	Description of Topic	Contact hours	C-D-I-O	IOs	Reference
	UNIT I–FUNDAMENTALS OF SOFTWARE MAINTENANCE	**9**			
1.	Meaning of software maintenance, software change, ongoing support, economic implications of modifying software, the nomenclature and image problem	1	C	1-4	1
2.	Software maintenance framework, potential solutions to maintenance problem.	2	C	1	1
3.	Maintenance process models :Definition of critical appraisal of traditional process models, maintenance process models.	2	C,D	1	1,2
4.	Program understanding: Aims of program comprehension, maintainers and their information needs comprehension process models	2	C,D	1	1
5.	Mental models, program comprehension strategies, factors that affect understanding, implications of comprehension theories and studies	2	C,D	1	1,2
	UNIT II-REVERSE ENGINEERING	**9**			
6.	Definition, purposes and objectives, levels of reverse Engineering, supports techniques, benefits.	3	C	2,3	1,3
7.	Reuse and reusability: Definitions, objective and benefit of reuse, approach to reuse,	3	C,D	3,4	1,3,4
8.	Domain Analysis, Components engineering, reuse process model, factors that impact upon reuse.	2	D,I	2,3,4	1,3,4
9.	Maintenance measures, Definitions, objectives of software measurement, example measures, guidelines for selecting maintenance measures	1	C,D	1	1
	UNIT III-CONFIGURATION MANAGEMENT	**9**			
10.	Definition for configuration management, change control, documentation	2	C	3	1
11.	Management and organizational issues, Management	2	C,D,I	3	1,

	responsibilities,				
12.	Enhancing maintenance productivity, maintenance teams	2	C,D	3	1
13.	Personnel Education and Training	2	C,I	2	1,3
14.	Organization modes	1	C	1	1
	UNIT IV-BUILDING AND SUSTAINING MAINTAINABILITY	**9**			
15.	Quality Assurance, fourth generation languages, object oriented paradigms	3	C	3	1
16.	Maintenance tools: Criteria for selecting tools, taxonomy of tools,	3	D,I,O	4	1
17.	Program understanding and reverse engineering testing, configuration management, and other tasks.	2	D,I	4	1
18.	Past, present and future of software maintenance	1	C	1	1
	UNIT V-SOFTWARE ADMINISTRATION	**9**			
19.	Analyzing system logs, operating system updates, patches, and configuration changes	2	D,I,O	5	4
20.	Performing backups. Installing and configuring new hardware and software.	1	D,I,O	5	4
21.	Adding, removing, or updating user account information, resetting passwords, System performance tuning	3	D,I,O	5	5
22.	Performing routine audits of systems and software	3	D,I,O	5	1
	Total contact hours	45			

LEARNING RESOURCES	
Sl. No.	**TEXT BOOKS**
1.	Armstrong A Takang and Penny A.Grubb, "Software Maintenance: concepts and Practice", International Thomson Computer press, London,2015
2.	Roger S Pressman, "Software Engineering", 6th edition, Tata McGraw-Hill, 2004
	REFERENCE BOOKS/OTHER READING MATERIAL
3.	Alain April, Alain Abran" Software Maintenance Management: Evaluation and Continuous Improvement ", Wiley- IEEE Computer Society Press, 2008.
4.	Analyzing System Logs: A New View of What's Important, IEEE Computer Society Press, 2007-
5.	Performance Tuning in Database Management System based on Analysis of Combination of Time and Cost Parameter through Neural Network Learning, International Journal of Computer Applications (0975 – 8887) Volume 96– No.1, June 2014

Course nature			**Theory**				
Assessment Method (Weightage 100%)							
In-semester	**Assessment tool**	Cycle test I	Cycle test II	Cycle Test III	Surprise Test	Quiz	**Total**
	Weightage	**10%**	**15%**	**15%**	**5%**	**5%**	**50%**
					End semester examination Weightage :		**50%**

15SE426E	**Software Measurements And Metrics**	**L**	**T**	**P**	**C**
		3	**0**	**0**	**3**

Co-requisite:	NIL
Prerequisite:	15SE314J
Data Book / Codes/Standards	NIL
Course Category	P PROFESSIONAL ELECTIVE
Course designed by	Department of Software Engineering
Approval	32nd Academic Council Meeting, 23rd July, 2016

PURPOSE	To Acquire Knowledge about Software Measurements and Metrics and apply the metrics for managing projects and decision making.

INSTRUCTIONAL OBJECTIVES		**STUDENT OUTCOMES**						
At the end of the course, student will be able to								
1.	Familiarize with different metrics used in different process levels	a	b					
2.	Apply metrics knowledge to measure Engineering problems.	a	b	e				
3.	Improve their ability in making decisions via continuous practice like assessment and usage of metrics.	a	h	i				
4.	Design, implement and change metrics based on industry practice	a	h	i				

Session	**Description of Topic**	**Contact hours**	**C-D-I-O**	**IOs**	**Reference**
	UNIT I: SOFTWARE SIZING METRICS	10			
1.	Fundamentals in Measurement - Basic Measures	1	C	1	1
2.	Evolution of software measurements ,	1	C	1	2
3.	Size Metrics: Lines of Code - Functional Size Measurements	4	C,I	1,2	2
4.	Software measures and metrics not based on function points	1	C	1,4	2
5.	Future Technical Developments in Functional Metrics	1	C	1	2
6.	Case Study to Measure software size using various size Metrics	2	C, O	1,2	1,2
	UNIT II: **COMPLEXITY AND TESTING METRICS**	9			
7.	Complexity Metrics and Models :Halstead's Software Science, Cyclomatic Complexity, Syntactic Constructs, Structure Metrics	3	C,I	1	1
8.	**Case Study for the usage of complexity metrics with tools**	1	C,O	1,2	1
9.	Testing Metrics :Test Progress S Curve - Testing Defect Arrivals Over Time - Product Size Over Time - CPU Utilization	3	C,I	1	1
10.	Effort/Outcome Model.	1	C	1	1
11.	**Case Study to apply Testing metrics**	1	I	1,2	1
	UNIT III : CUSTOMER SATISFACTION ANALYSIS AND OBJECT ORIENTED METRICS	8			
12.	Customer Satisfaction Surveys: Methods of survey data collection - Sampling Methods	2	C,D	1,3	1,2
13.	Analyzing Satisfaction with practice	2	C,D	1,3	1,2
14.	Metrics for Object-Oriented Projects with tools	2	C,I	1,2	1,3,4,5
15.	Lorenz Metrics, CKOO Metrics Suite	2	C,I	1,2	1,3,4,5
	UNIT IV: SOFTWARE QUALITY METRICS	10			
16.	Software Quality - Five steps to software quality control	1	C	3	1
17.	Product Quality Metrics with practice	2	C	3	2
18.	In-Process Quality Metrics with practice	2	C,O	1,2	1
19.	Metrics for Software Maintenance	1	C,O	1,2	1
20.	Ishikawa's Seven Basic Tools with practice	2	C,O	1,2	1
21.	Defect Removal Effectiveness with practice	2	C,I	1,2	1,2

	UNIT V: ASSESSMENTS AND PROCESS IMPROVEMENT	8			
22.	Conducting In-Process Quality Assessments	2	C	1,3	1,2
23.	Conducting Software Project Assessments	2	C,O	1,3	1,2
24.	Measures and metrics of industry leaders	2	C,O	1,3	1,2
25.	Software Process Improvement Sequences	2	C	3,4	1
	Total contact hours	45			

LEARNING RESOURCES	
Sl. No.	**TEXT BOOKS**
1.	Stephen H. Kan, "Metrics and Models in Software Quality Engineering", Addison Wesley, 2011.
2.	Caper Jones, "Applied Software Measurement: Global Analysis of Productivity and Quality", Third Edition, McGraw Hill Companies, 2008
	REFERENCE BOOKS/OTHER READING MATERIAL
3.	Mark Lorenz, Jeff Kidd, "Object-Oriented Software Metrics", Prentice Hall, 2000
4.	Naresh Chauhan, "Software Testing Principles and Practices", Oxford University Press, 2010.
5.	RavindranathPandian C, "Software Metrics A Guide to planning, Analysis, and Application", Auerbach, First Indian Reprint, 2011

Course nature		**Theory**					
Assessment Method (Weightage 100%)							
In-semester	**Assessment tool**	Cycle test I	Cycle test II	Cycle Test III	Surprise Test	Quiz	**Total**
	Weightage	**10%**	**15%**	**15%**	**5%**	**5%**	**50%**
					End semester examination Weightage :		**50%**

15SE427E	Wireless And Mobile Communication	L	T	P	C
		3	0	0	3
Co-requisite:	NIL				
Prerequisite:	NIL				
Data Book / Codes/Standards	NIL				
Course Category	P | PROFESSIONAL ELECTIVE				
Course designed by	Department of Software Engineering				
Approval	32nd Academic Council Meeting, 23rd July, 2016				

PURPOSE	The purpose of learning this course is to know about the fundamental concepts of mobile communications and wireless networks technologies.							
INSTRUCTIONAL OBJECTIVES		**STUDENT OUTCOMES**						
At the end of the course, student will be able to								
1.	Apply wireless technology concepts to Engineering problems related to communication	a						
2.	Improve their knowledge on digital and analog modulation techniques.	a	c					
3.	Equip themselves familiar with principles of mobile communication	a						
4.	Familiarize with the digital cellular standards.	a	c					
5.	Expose to the emerging wireless technologies	a	b					

Session	Description of Topic	Contact hours	C-D-I-O	IOs	Reference
	UNIT I: INTRODUCTION TO ANALOG AND DIGITAL MODULATION SCHEMES	**9**			
1.	Introduction to wireless communication and Elements of a wireless communication system, signal and noise	2	C	1-2	1,2,3,4,5
2.	The radio _frequency spectrum- Analog modulation schemes - Amplitude modulation- frequency and phase modulation	3	C	2	1,2,3,4
3.	Introduction to Digital modulation - Frequency shift keying- Phase shift keying	2	C,D	2	1,2,3,4
4.	Multiplexing and Multiple access- Spread spectrum systems.	2	C,D	2	1,2,3,4
	UNIT II: PRINCIPLES OF MOBILE COMMUNICATION	**9**			
5.	Cellular concept- Cell area- signal strength and cell parameter- capacity of cell	2	C	2	1,3,4,6
6.	Co channel interference-Frequency reuse concept- Cell splitting – cell sectoring-multiple radio access protocols	2	C,D	2	1,2,3
7.	Frequency division Multiple Access – Time Division Multiple Access- Space Division Multiple Access-Code Division Multiple Access	3	D	2	2
8.	OFDM-Comparison of multiple division techniques.	2	D,I	1,2	2
	UNIT III: DIGITAL CELLULAR STANDARDS	**9**			
9.	GSM -frequency bands and channels- frames in GSM – Interfaces ,planes and layers of GSM	2	C	4	2,7
10.	Handoff-short messaging service-GPRS-EDGE-	2	D	4	1,2
11.	3G CELLULAR SYSTEMS-MMS-UMTS-Satellite system infrastructure-GPS-Limitations of GPS-Beneficiaries of GPS-	3	C.D	4	2,9
12.	4G cellular systems - **4G –standards(LTE/wimax)**	2	C,D	4	5,6,9

	UNIT IV: MOBILE NETWORK AND TRANSPORT LAYER	**9**			
13.	Mobile IP-goals-IP Packet delivery-Tunneling-Reverse tunneling	2	C,D	3	3,9
14.	IPV6-Dynamic host routing protocol	1	D	3	6
15.	Traditional TCP-Congestion control-classical TCP-Snooping-Mobile TCP	2	D	3	3
16.	Transaction oriented TCP-TCP over 2.5/3G Wireless Networks	2	C	3	3
17.	Wireless Application protocol-architecture-wireless transport layer security-wireless markup language-Push architecture.	2	C,D	3	3
	UNIT V:EMERGING WIRELESS TECHNOLOGIES	**9**			
18.	IEEE 802.11-IEEE802.15, Mobile ad hoc networks – Characteristics	2	D,I	5	2
19.	Routing-AODV ,VANETS - wireless sensor networks – RFID technology	3	D,I	5	2,8
20.	Wi –Fi standards – Wimax standards	2	C,D	5	2,8
21.	Femtocell network – Push -to –talk technology for SMS.	2	D,I	5	2
	Total contact hours	45			

LEARNING RESOURCES	
Sl. No.	**TEXT BOOKS**
1.	Roy Blake, "Wireless communication technology" CENGAGE Learning, sixth Indian reprint 2013.
2.	Dharma PrakashAgrawal , Qing –An Zeng , " Introduction to wireless and mobile systems" CENGAGE Learning, first edition 2014.
3.	Jochen Schiller, "Mobile Communications", Addision Wesley, 2nd Edition, 2011.
4.	Singal T.L. , "Wireless communication" Tata McGraw Hill Education private limited , 2011.
	REFERENCE BOOKS/OTHER READING MATERIAL
5.	G.I Papadimitriou, A.S. Pomportsis, P.Nicopolitidis, M.S.Obaidat, "Wireless Networks", John-Wiley and Sons,2003.
6.	Gray J.Mullet,"Wireless Telecommunications Systems And Networks ",CengageLearning,Reprint 2014.
7.	UpenaDalal, "Wireless communication" Oxford University press, first edition 2009.
8.	KavehPahlavan&Prashant Krishnamurthy, "Wireless Networks" PHI,2002.
9.	MartynMallick, "Mobile and Wireless Design Essentials", Wiley Dreamtech India Pvt. Ltd., 2004.

Course nature		**Theory**					
Assessment Method (Weightage 100%)							
In-semester	**Assessment tool**	Cycle test I	Cycle test II	Cycle Test III	Surprise Test	Quiz	**Total**
	Weightage	**10%**	**15%**	**15%**	**5%**	**5%**	**50%**
					End semester examination Weightage :		**50%**

15SE251E	Principles Of Object Oriented Programming	L	T	P	C
		3	0	0	3
Co-requisite:	NIL				
Prerequisite:	NIL				
Data Book / Codes/Standards	NIL				
Course Category	**P** PROFESSIONAL ELECTIVE				
Course designed by	Department of Software Engineering				
Approval	32nd Academic Council Meeting, March, 2016				

PURPOSE	Real world is full of objects. The worldly problems can be best solved using object oriented approach. The pioneer programming language to implement object oriented features is C++. This course aims at building object oriented skills through programming in C++.							
INSTRUCTIONAL OBJECTIVES		**STUDENT OUTCOMES**						
At the end of the course, student will be able to								
1.	Apply the Basic Object Oriented Features	a	b					
2.	Develop Generic programming	a	b					
3.	Decide, apply appropriate data structures and solve problems by applying the skills acquired so far	a	e	k				
4.	Break a programming problem to Classes, Design and Develop a full working code	a	b					
5.	Develop programs using Streams, Files, Templates and handle Exceptions	a	b					

Session	Description of Topic	Contact hours	C-D-I-O	IOs	Reference
	UNIT I: PRINCIPLES OF OBJECT-ORIENTED PROGRAMMING	**10**			
1.	Need of Object-Oriented Programming - Comparison of procedure-Oriented Programming and Object-Oriented Programming	1	C	1	1
2.	Basic Concepts of Object-Oriented Programming	1	C	1	1
3.	Benefits of OOP, Characteristics of Object-Oriented Languages, Applications of OOP	1	C	1	1
4.	C++ Programming Basics: Basic Program Construction, Simple C++ Program	1	C	1,2	1
5.	Tokens	1	C	1,2	1
6.	Data Types, Type Compatibility	2	C	1,2	1
7.	Storage Classes, Symbolic Constants	1	C	1,2	1
8.	Variables, Dynamic Initialization of Variables, References Variables	1	C	1,2	1
9.	Operators, Manipulators and Operator Precedence, Expressions	1	C	1,2	1
	UNIT II: CONTROL STRUCTURES AND FUNCTIONS	**9**			
10.	Loops and Decisions	2	C,I	1,2	1
11.	Arrays and Structures	2	C,I	1,2	1
12.	Strings	1	C,I	1,2	1
13.	Functions – Prototyping, Simple Functions, Passing arguments, Returning values	2	C,I	1,2	1
14.	Default Arguments, Reference Arguments, Recursion	1	C,I	1,2	1
15.	Inline Functions, Math Library Fuctions	1	C,I	1,2	1
	UNIT III: FEATURES OF OBJECT-ORIENTED PROGRAMMING	**9**			
16.	Introduction to Classes and Objects	1	C	1,2	1,2
17.	Member Functions and Nesting of Member Functions	1	C,I	1,4	1,2
18.	Memory Allocation for Objects, Static Members	1	C,I	1,2	1,2
19.	Array of Objects, Objects as Function Arguments	1	C,I	1,3,4	1,2
20.	Friend Function	1	C,I	1,2	1,2
21.	Constructors and its types, Destructors	2	C,I	1,4	1,2

22.	Function Overloading	1	C,I	1,3,4	1,2
23.	Operator Overloading – Overloading Unary and Binary Operators	1	C,I	1,3,4	1,2
	UNIT IV: INHERITANCE AND POINTERS	**7**			
24.	Inheritance, Defining Derived Classes	1	C,I	1,4	1,2
25.	Types of Inheritance	2	C,I	1,3,4	1,2
26.	Virtual Base Classes and Abstract Classes	1	C,I	1,4	1
27.	Pointers, Pointers with Arrays and Functions	2	C,I	1,3,4	1,2
28.	Virtual Functions	1	C,I	1,3,4	1
	UNIT III: STREAMS AND FILES, TEMPLATES, EXCEPTIONS	**10**			
29.	Streams: Classes and Errors	2	C	5	2
30.	Disk File I/O with Streams	2	C,I	5	2
31.	Files: File Pointers	1	C	5	2
32.	Templates – Function templates, Class templates	2	C,I	5	1,2
33.	Exceptions: Need of Exceptions, keywords, Simple and Multiple Exceptions	1	C,I	5	1,2
34.	Re-throwing Exception and Exception Specifications, Custom Exception	2	C,I	5	1,2
	Total contact hours	45			

LEARNING RESOURCES	
Sl. No.	**TEXT BOOKS**
1.	E.Balaguruswamy "Object Oriented Programming with C++", 6th Edition, Tata McGraw Hill Education, 2015
2.	Robert Lafore, "Object-Oriented Programming in C++", 4th Edition, SAMS Publishing, 2008
	REFERENCE BOOKS/OTHER READING MATERIAL
3.	BjarneStroustrup ,"The C++ Programming Language", 4th Edition, Addison Wesley, 2015
4.	Deitel, "C++ How to Program", 6th edition, PHI publication, 2008
5.	Bhusan Trivedi, "Programming with ANSI C++", 2nd edition, Oxford higher education,2014

Course nature				**Theory**			
Assessment Method (Weightage 100%)							
In-semester	**Assessment tool**	Cycle test I	Cycle test II	Cycle Test III	Surprise Test	Quiz	**Total**
	Weightage	**10%**	**15%**	**15%**	**5%**	**5%**	**50%**
					End semester examination Weightage :		**50%**