

**ANNA UNIVERSITY, CHENNAI**  
**AFFILIATED INSTITUTIONS**  
**M.E. SOFTWARE ENGINEERING**  
**REGULATIONS – 2017**  
**CHOICE BASED CREDIT SYSTEM**

**PROGRAM EDUCATIONAL OBJECTIVES (PEOS):**

1. Graduates will have a successful career in software industries, research and higher education.
2. Pursue life-long learning and will have the ability for adapting to evolving technological advancements

**PROGRAM OUTCOMES (POS):**

**Engineering Graduates will be able to:**

1. Apply knowledge of computing, mathematics, science and engineering for software systems
2. Demonstrate a basic understanding of software engineering practices from vision to analysis, design, development, validation, deployment and maintenance.
3. Identify and solve complex engineering problems and tasks using software engineering principles and methodologies with appropriate consideration for public health and safety, cultural, societal and environmental considerations.
4. Develop skills to create and use various software Engineering based techniques and tools to solve complex Engineering problems.
5. Function effectively as an individual, and as a member or leader in diverse teams and in multidisciplinary system.
6. Understand the best software practices and processes for a contemporary application domain and able to take decisions based on real world examples.
7. Demonstrate knowledge and understanding of effective management of software projects.
8. Conduct investigations for complex problems to get a broader perspective of the discipline through research by designing and conducting experiments as well as analyzing and interpreting the results.
9. Understand, reflect and commit to professional ethics and norms of software engineering practice for sustainable development of society .
10. Progress successfully in their profession and recognize the importance of self-education and life-long learning.
11. Communicate effectively on complex software engineering activities with domain-experts as well as non-experts through effective presentations, written forms and constructive documentation.

**Program Specific Objectives (PSOs)**

Graduates of M.E. Software Engineering will be able to

- 1) Apply software engineering principles and practices for designing and testing quality software and for scientific and business applications.
- 2) Adapt to emerging information and communication technologies to innovate ideas to solve the societal problems.
- 3) Analyze the real world problem to get a broader perspective of the discipline through research

## MAPPING OF POS/PSOS TO PEOS FOR M.E. SOFTWARE ENGINEERING

POs	PEOs	
	Graduates will have a successful career in software research and higher education.	Pursue life-long learning and will have the ability for adapting to evolving technological advancements.
1. Apply knowledge of computing, mathematics, science and engineering for software systems	<b>3</b>	<b>1</b>
2. Demonstrate a basic understanding of software engineering practices from vision to analysis, design, development, validation, deployment and maintenance.	<b>3</b>	<b>2</b>
3. Identify and solve complex engineering problems and tasks using software engineering principles and methodologies with appropriate consideration for public health and safety, cultural, societal and environmental considerations	<b>3</b>	<b>2</b>
4. Develop skills to create and use various software Engineering based techniques and tools to solve complex Engineering problems	<b>3</b>	<b>2</b>
5. Function effectively as an individual, and as a member or leader in diverse teams and in multidisciplinary system	<b>3</b>	<b>2</b>
6. Understand the best software practices and processes for a contemporary application domain and able to take decisions based on real world examples	<b>2</b>	<b>2</b>
7. Demonstrate knowledge and understanding of effective management of software projects	<b>2</b>	<b>2</b>
8. Conduct investigations for complex problems to get a broader perspective of the discipline through research by designing and conducting experiments as well as analyzing and interpreting the results	<b>3</b>	<b>2</b>
9. Understand, reflect and commit to professional ethics and norms of software engineering practice for sustainable development of society	<b>3</b>	<b>2</b>

10. Progress successfully in their profession and recognize the importance of self-education and life-long learning	<b>1</b>	<b>3</b>
11. Communicate effectively on complex software engineering activities with domain-experts as well as non-experts through effective presentations, written forms and constructive documentation	<b>3</b>	<b>3</b>
<b>PSOs</b>		
1. Apply software engineering principles and practices for designing and testing quality software and for scientific and business applications.	<b>3</b>	<b>2</b>
2. Adapt to emerging information and communication technologies (ICT) to innovate ideas to solve the societal problems.	<b>1</b>	<b>3</b>
3. Analyze the real world problem to get a broader perspective of the discipline through research	<b>3</b>	<b>2</b>

**Contribution**

**1: Reasonable**

**2: Significant**

**3: Strong**

**M.E. SOFTWARE ENGINEERING  
SEMESTER COURSE WISE PO MAPPING**

		SUBJECTS	PROGRAMME OUTCOMES											
			PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
I Y E A R	SEMESTER I	Applied Probability and Statistics	3	1	1	1	2	1	1	2	1	1	1	2
		Advanced Data Structures and Algorithms	3	1	3	2	1	1	1	2	2	2	1	2
		Software Architecture	3	2	3	3	3	2	3	2	2	1	2	2
		Advanced Software Engineering	2	2	3	1	3	1	2	3	1	1	1	1
		Software Requirements Engineering	2	2	2	1	2	1	1	1	2	3	2	2
		Data Structures Laboratory	3	1	3	1	1	1	1	2	3	2	2	2
		Term Paper Writing and Seminar	1	3	3	3	3	3	2	3	3	3	1	3
II Y E A R	SEMESTER II	Software Testing Methodologies and Quality Assurance	1	1	1	2	3	3	2	3	1	1	1	2
		Software Industrialization	3	1	2	2	2	1	2	1	1	1	2	1
		Software Reliability Metrics and Models	1	1	2	3	3	1	1	2	1	2	2	1
		Advanced Databases	2	1	2	2	3	2	3	2	1	1	1	1
		<b>Professional Elective I</b>												
		Agile Software Engineering	2	2	3	2	2	3	2	2	1	1	1	3
		Business Intelligence	1	2	3	2	3	3	3	2	1	1	1	2
		Cloud Computing Technologies	2	1	2	2	3	2	2	2	1	1	1	2
		Software Test Automation	1	1	1	2	3	1	1	1	1	1	1	2
		Big Data Analytics	2	3	3	2	3	2	1	3	1	1	1	2

1 Y E A R	SEMESTER II	<b>Professional Elective II</b>												
		Knowledge Management	3	1	3	3	1	2	1	2	1	1	2	2
		<u>Software Verification and Validation</u>	1	2	1	3	3	1	1	2	1	1	1	2
		<u>Machine Learning Techniques</u>	2	2	3	2	3	1	1	1	1	1	1	2
		<u>Virtualization Techniques and Applications</u>	2	3	2	1	3	1	3	2	1	1	1	2
		<u>Internet of Things</u>	3	2	3	2	2	3	3	2	1	1	2	2
		<u>Software Development Lab</u>	3	3	3	3	3	1	1	2	3	3	2	2
	SEMESTER III	Integrated Software Project Management	3	2	3	2	3	3	2	2	1	3	3	2
		Project Work Phase – I	3	3	3	3	3	3	2	3	3	3	2	3
		<b>Professional Elective –III</b>												
		<u>Web Design and Management</u>	1	2	3	1	3	3	3	3	1	1	1	3
		<u>Social Network Mining and Analysis</u>	3	3	3	3	3	2	1	2	1	1	2	2
		<u>Test Driven Development</u>	3	3	3	3	3	2	1	2	1	1	1	2
		<u>Personal Software Process</u>	1	1	1	1	3	2	1	2	1	2	3	2
		Software Security	1	2	2	1	3	1	1	2	1	1	1	2
		<b>Professional Elective IV</b>												
		<u>Enterprise Application Integration</u>	3	3	3	1	2	1	1	1	1	1	1	2
		<u>Managing Human Resource</u>	-	2	3	1	1	3	3	2	1	2	2	2
		<u>Principles of Supply Chain Management</u>	2	2	2	2	2	3	3	2	1	1	3	2
		<u>Software Agents</u>	2	2	3	2	2	2	1	2	1	1	2	2
<u>User Interface Design and Evaluation</u>	1	1	2	1	3	3	2	1	1	2	1	2		
<b>Professional Elective V</b>														
<u>Service Oriented Architecture</u>	3	2	3	2	3	2	2	2	1	1	2	2		
Real Time Systems	2	2	1	2	2	1	1	2	1	2	1	1		
<u>Security Practices</u>	3	1	1	2	2	1	1	1	2	1	1	1		
<u>Information Retrieval Techniques</u>	2	2	2	3	3	2	2	2	1	1	1	2		

	Embedded System Design													
<b>SEM IV</b>														
	Project Work Phase – II													

**1: REASONABLE 2: SIGNIFICANT 3: STRONG**

**ANNA UNIVERSITY, CHENNAI**  
**AFFILIATED INSTITUTIONS**  
**M.E. SOFTWARE ENGINEERING**  
**REGULATIONS – 2017**  
**CHOICE BASED CREDIT SYSTEM**  
**I - IV SEMESTERS CURRICULA AND SYLLABI**

**SEMESTER I**

SL. NO	COURSE CODE	COURSE TITLE	CATEGORY	CONTACT PERIODS	L	T	P	C
<b>THEORY</b>								
1.	MA5160	Applied Probability and Statistics	FC	4	4	0	0	4
2.	CP5151	Advanced Data Structures and Algorithms	PC	4	4	0	0	4
3.	SE5101	Software Architecture	PC	3	3	0	0	3
4.	CP5154	Advanced Software Engineering	PC	3	3	0	0	3
5.	SE5102	Software Requirements Engineering	PC	3	3	0	0	3
<b>PRACTICALS</b>								
6.	CP5161	Data Structures Laboratory	PC	4	0	0	4	2
7.	CP5281	Term Paper Writing and Seminar	EEC	2	0	0	2	1
<b>TOTAL</b>				<b>23</b>	<b>17</b>	<b>0</b>	<b>6</b>	<b>20</b>

**SEMESTER II**

SL. NO	COURSE CODE	COURSE TITLE	CATEGORY	CONTACT PERIODS	L	T	P	C
<b>THEORY</b>								
1.	SE5201	Software Testing Methodologies and Quality Assurance	PC	5	3	0	2	4
2.	IF5251	Software Industrialization	PC	3	3	0	0	3
3.	SE5202	Software Reliability Metrics and Models	PC	3	3	0	0	3
4.	IF5191	Advanced Databases	PC	3	3	0	0	3
5.		Professional Elective I	PE	3	3	0	0	3
6.		Professional Elective II	PE	3	3	0	0	3
<b>PRACTICALS</b>								
7.	SE5211	Software Development Laboratory	PC	4	0	0	4	2
<b>TOTAL</b>				<b>24</b>	<b>18</b>	<b>0</b>	<b>6</b>	<b>21</b>

**SEMESTER III**

SL. NO	COURSE CODE	COURSE TITLE	CATEGORY	CONTACT PERIODS	L	T	P	C
<b>THEORY</b>								
1.	SE5301	Integrated Software Project Management	PC	3	3	0	0	3
2.		Professional Elective III	PE	3	3	0	0	3
3.		Professional Elective - IV	PE	3	3	0	0	3
4.		Professional Elective -V	PE	3	3	0	0	3
<b>PRACTICALS</b>								
5.	SE5311	Project Work Phase - I	EEC	12	0	0	12	6
<b>TOTAL</b>				<b>24</b>	<b>12</b>	<b>0</b>	<b>12</b>	<b>18</b>

**SEMESTER IV**

SL. NO	COURSE CODE	COURSE TITLE	CATEGORY	CONTACT PERIODS	L	T	P	C
<b>PRACTICALS</b>								
1.	SE5411	Project Work Phase - II	EEC	24	0	0	24	12
<b>TOTAL</b>					<b>0</b>	<b>0</b>	<b>24</b>	<b>12</b>

**TOTAL NO. OF CREDITS:71**



**FOUNDATION COURSES (FC)**

SL. NO	COURSE CODE	COURSE TITLE	CATEGORY	CONTACT PERIODS	L	T	P	C
1.	MA5160	Applied Probability and Statistics	FC	4	4	0	0	4

**PROFESSIONAL CORE (PC)**

SL. NO	COURSE CODE	COURSE TITLE	CATEGORY	CONTACT PERIODS	L	T	P	C
1.	CP5151	Advanced Data Structures and Algorithms	PC	4	4	0	0	4
2.	SE5101	Software Architecture	PC	3	3	0	0	3
3.	CP5154	Advanced Software Engineering	PC	3	3	0	0	3
4.	SE5102	Software Requirements Engineering	PC	3	3	0	0	3
5.	CP5161	Data Structures Laboratory	PC	4	0	0	4	2
6.	SE5201	Software Testing Methodologies and Quality Assurance	PC	5	3	0	2	4
7.	IF5251	Software Industrialization	PC	3	3	0	0	3
8.	SE5202	Software Reliability Metrics and Models	PC	3	3	0	0	3
9.	IF5191	Advanced Databases	PC	3	3	0	0	3
10.	SE5211	Software Development Laboratory	PC	4	0	0	4	2
11.	SE5301	Integrated Software Project Management	PC	3	3	0	0	3

**EMPLOYABILITY ENHANCEMENT COURSE (EEC)**

SL. NO	COURSE CODE	COURSE TITLE	CATEGORY	CONTACT PERIODS	L	T	P	C
1.	CP5281	Term Paper Writing and Seminar	EEC	2	0	0	2	1
2.	SE5311	Project Work Phase – I	EEC	12	0	0	12	6
3.	SE5411	Project Work Phase - II	EEC	24	0	0	24	12

**PROFESSIONAL ELECTIVES (PE)\*  
SEMESTER II  
ELECTIVE I**

SL. NO	COURSE CODE	COURSE TITLE	CATEGORY	CONTACT PERIODS	L	T	P	C
1.	SE5001	Agile Software Engineering	PE	3	3	0	0	3
2.	SE5002	Business Intelligence	PE	3	3	0	0	3
3.	CP5092	Cloud Computing Technologies	PE	3	3	0	0	3
4.	SE5003	Software Test Automation	PE	3	3	0	0	3
5.	CP5293	Big Data Analytics	PE	3	3	0	0	3

**SEMESTER II  
ELECTIVE II**

SL. NO	COURSE CODE	COURSE TITLE	CATEGORY	CONTACT PERIODS	L	T	P	C
1.	SE5004	Knowledge Management	PE	3	3	0	0	3
2.	SE5005	Software Verification and Validation	PE	3	3	0	0	3
3.	CP5191	Machine Learning Techniques	PE	3	3	0	0	3
4.	SE5006	Virtualization Techniques and Applications	PE	3	3	0	0	3
5.	CP5292	Internet of Things	PE	3	3	0	0	3

**SEMESTER III  
ELECTIVE III**

SL. NO	COURSE CODE	COURSE TITLE	CATEGORY	CONTACT PERIODS	L	T	P	C
1.	SE5007	Web Design and Management	PE	3	3	0	0	3
2.	SE5008	Social Network Mining and Analysis	PE	3	3	0	0	3
3.	SE5009	Test Driven Development	PE	3	3	0	0	3
4.	SE5010	Personal Software Process	PE	3	3	0	0	3
5.	SE5011	Software Security	PE	3	3	0	0	3

**SEMESTER III  
ELECTIVE IV**

<b>SL. NO</b>	<b>COURSE CODE</b>	<b>COURSE TITLE</b>	<b>CATEGORY</b>	<b>CONTACT PERIODS</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
1.	SE5012	Enterprise Application Integration	PE	3	3	0	0	3
2.	SE5013	Managing Human Resource	PE	3	3	0	0	3
3.	SE5014	Principles of Supply Chain Management	PE	3	3	0	0	3
4.	SE5015	Software Agents	PE	3	3	0	0	3
5.	SE5016	User Interface Design and Evaluation	PE	3	3	0	0	3

**SEMESTER III  
ELECTIVE V**

<b>SL. NO</b>	<b>COURSE CODE</b>	<b>COURSE TITLE</b>	<b>CATEGORY</b>	<b>CONTACT PERIODS</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
1.	SE5091	Service Oriented Architecture and Design	PE	3	3	0	0	3
2.	MP5291	Real Time Systems	PE	3	3	0	0	3
3.	CP5291	Security Practices	PE	3	3	0	0	3
4.	CP5094	Information Retrieval Techniques	PE	3	3	0	0	3
5.	AP5191	Embedded System Design	PE	3	3	0	0	3



**OBJECTIVES:**

- To understand the usage of algorithms in computing.
- To learn and use hierarchical data structures and its operations
- To learn the usage of graphs and its applications.
- To select and design data structures and algorithms that is appropriate for problems.  
To study about NP Completeness of problems.

**UNIT I      ROLE OF ALGORITHMS IN COMPUTING****12**

Algorithms – Algorithms as a Technology- Insertion Sort – Analyzing Algorithms – Designing Algorithms- Growth of Functions: Asymptotic Notation – Standard Notations and Common Functions- Recurrences: The Substitution Method – The Recursion-Tree Method

**UNIT II      HIERARCHICAL DATA STRUCTURES****12**

Binary Search Trees: Basics – Querying a Binary search tree – Insertion and Deletion- Red-Black trees: Properties of Red-Black Trees – Rotations – Insertion – Deletion -B-Trees: Definition of B-trees – Basic operations on B-Trees – Deleting a key from a B-Tree- Fibonacci Heaps: structure – Mergeable-heap operations- Decreasing a key and deleting a node-Bounding the maximum degree.

**UNIT III      GRAPHS****12**

Elementary Graph Algorithms: Representations of Graphs – Breadth-First Search – Depth-First Search – Topological Sort – Strongly Connected Components- Minimum Spanning Trees: Growing a Minimum Spanning Tree – Kruskal and Prim- Single-Source Shortest Paths: The Bellman-Ford algorithm – Single-Source Shortest paths in Directed Acyclic Graphs – Dijkstra’s Algorithm; All-Pairs Shortest Paths: Shortest Paths and Matrix Multiplication – The Floyd-Warshall Algorithm;

**UNIT IV      ALGORITHM DESIGN TECHNIQUES****12**

Dynamic Programming: Matrix-Chain Multiplication – Elements of Dynamic Programming – Longest Common Subsequence- Greedy Algorithms: An Activity-Selection Problem – Elements of the Greedy Strategy- Huffman Codes.

**UNIT V      NP COMPLETE AND NP HARD****12**

NP-Completeness: Polynomial Time – Polynomial-Time Verification – NP- Completeness and Reducability – NP-Completeness Proofs – NP-Complete Problems

**TOTAL: 60 PERIODS****OUTCOMES:****Upon the completion of the course the students should be able to:**

- Design data structures and algorithms to solve computing problems.
- Design algorithms using graph structure and various string matching algorithms to solve real-life problems.
- Apply suitable design strategy for problem solving

**REFERENCES:**

1. Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, “Data Structures and Algorithms”, Pearson Education, Reprint 2006.
2. Robert Sedgewick and Kevin Wayne, “ALGORITHMS”, Fourth Edition, Pearson Education.
3. S.Sridhar, “Design and Analysis of Algorithms”, First Edition, Oxford University Press. 2014
4. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, “Introduction to Algorithms”, Third Edition, Prentice-Hall, 2011.

SE5101

**SOFTWARE ARCHITECTURE**

L	T	P	C
3	0	0	3

**OBJECTIVES:**

**The student should be able to**

- Understand the fundamentals of software architecture.
- Study the various software development methodologies.
- Learn the various software architecture design components.
- Relate software architecture and software quality.

**UNIT I INTRODUCTION**

**9**

Basic Concepts of Software Architecture - Architecture business cycle - architectural patterns - reference models - architectural structures, views - Introduction to Styles - Simple Styles - Distributed and Networked Architectures-Architecture for network based applications - Decentralized Architectures.

**UNIT II DESIGN METHODOLOGIES**

**9**

Structured Design - Design Practices – Stepwise Refinement – Incremental Design – Structured System Analysis and Design – Jackson Structured Programming – Jackson System Development

**UNIT III ARCHITECTURE DESCRIPTION, DOCUMENTATION AND EVALUATION**

**9**

Early Architecture Description Languages –Domain and Style Specific ADLs –Extensible ADLs - Documenting Software architecture -Architecture Evaluation –ATAM.

**UNIT IV ARCHITECTURE DESIGN**

**9**

Typical Architectural Design - Data Flow - Independent Components - Call and Return - Using Styles in Design – choices of styles – Architectural design space – Theory of Design Spaces – Design space of Architectural Elements – Design space of Architectural styles.

**UNIT V CREATING AN ARCHITECTURE**

**9**

Understanding Quality Attributes - Functionality and Architecture –Architecture and Quality Attributes-System Quality Attributes –Quality attribute Scenarios in Practice - Introducing Tactics - Availability Tactics –Modifiability Tactics –Performance Tactics -Security Tactics –Testability Tactics –Usability Tactics –Relationship of Tactics to Architectural Patterns –Architectural Patterns and Styles.

**TOTAL : 45 PERIODS**

**OUTCOMES:**

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

- Develop Software applications starting from software architecture and design.
- Learn and evaluate existing software architectures.
- Design methods for improving software quality from the perspective of software architecture.

**REFERENCES:**

1. David Budgen, "Software Design", Second Edition, Pearson Education, 2004.
2. Hong Zhu, "Software Design Methodology from Principles to Architectural Styles", Elsevier, 2005.
3. Len Bass, Paul Clements, Rick Kazman, "Software Architecture in Practice, Third Edition, Addison,Wesley, 2012.
4. Mary shaw and David Garlan, Software Architecture –Perspectives on an emerging discipline, Pearson education, 2008.
5. Richard N.Taylor, NenadMedvidovic and Eric M.Dashofy, "Software Architecture, Foundations, Theory and Practice", Wiley 2010.

**OBJECTIVES:**

- To understand Software Engineering Lifecycle Models
- To do project management and cost estimation
- To gain knowledge of the System Analysis and Design concepts.
- To understand software testing approaches
- To be familiar with DevOps practices

**UNIT I INTRODUCTION****9**

Software engineering concepts – Development activities – Software lifecycle models - Classical waterfall - Iterative waterfall – Prototyping – Evolutionary - Spiral – Software project management – Project planning – Estimation – Scheduling – Risk management – Software configuration management.

**UNIT II SOFTWARE REQUIREMENT SPECIFICATION****9**

Requirement analysis and specification – Requirements gathering and analysis – Software Requirement Specification – Formal system specification – Finite State Machines – Petrinets – Object modelling using UML – Use case Model – Class diagrams – Interaction diagrams – Activity diagrams – State chart diagrams – Functional modelling – Data Flow Diagram.

**UNIT III ARCHITECTURE AND DESIGN****9**

Software design – Design process – Design concepts – Coupling – Cohesion – Functional independence – Design patterns – Model-view-controller – Publish-subscribe – Adapter – Command – Strategy – Observer – Proxy – Facade – Architectural styles – Layered - Client-server - Tiered - Pipe and filter.- User interface design

**UNIT IV TESTING****9**

Testing – Unit testing – Black box testing– White box testing – Integration and System testing– Regression testing – Debugging - Program analysis – Symbolic execution – Model Checking

**UNIT V DevOps****9**

DevOps:Motivation-Cloud as a platform-Operations- Deployment Pipeline:Overall Architecture-Building and Testing-Deployment- Case study: Migrating to Microservices.

**TOTAL: 45 PERIODS****OUTCOMES:**

**At the end of this course, the students will be able to:**

- Understand the advantages of various Software Development Lifecycle Models
- Gain knowledge on project management approaches as well as cost and schedule estimation strategies
- Perform formal analysis on specifications
- Use UML diagrams for analysis and design
- Architect and design using architectural styles and design patterns
- Understand software testing approaches
- Understand the advantages of DevOps practices

## REFERENCES:

1. Bernd Bruegge, Alan H Dutoit, Object-Oriented Software Engineering, 2<sup>nd</sup> edition, Pearson Education, 2004.
2. Carlo Ghezzi, Mehdi Jazayeri, Dino Mandrioli, Fundamentals of Software Engineering, 2<sup>nd</sup> edition, PHI Learning Pvt. Ltd., 2010.
3. Craig Larman, Applying UML and Patterns, 3rd ed, Pearson Education, 2005.
4. Len Bass, Ingo Weber and Liming Zhu, “DevOps: A Software Architect’s Perspective”, Pearson Education, 2016
5. Rajib Mall, Fundamentals of Software Engineering, 3<sup>rd</sup> edition, PHI Learning Pvt. Ltd., 2009.
6. Stephen Schach, Software Engineering 7th ed, McGraw-Hill, 2007.

SE5102

SOFTWARE REQUIREMENTS ENGINEERING

L T P C  
3 0 0 3

## OBJECTIVES

The student should be able to

- Understand the basics of requirements engineering
- Learn different techniques used for requirements elicitation
- Know the role played by requirements analysis in requirement integration
- Appreciate the use of various methodologies for requirements development
- Study the current trends in requirements prioritization and validation.

### UNIT I REQUIREMENTS ENGINEERING OVERVIEW 9

Software Requirement Overview – Software Development Roles –Software Development Process Kernels – Commercial Life Cycle Model – Vision Development – Stakeholders Needs & Analysis – Stakeholder needs –Stakeholder activities.

### UNIT II REQUIREMENTS ELICITATION 9

The Process of Requirements Elicitation – Requirements Elicitation Problems – Problems of Scope – Problems of Understanding – Problems of Volatility – Current Elicitation Techniques – Information Gathering – Requirements Expression and Analysis – Validation – An Elicitation Methodology Framework – A Requirements Elicitation Process Model – Methodology over Method – Integration of Techniques – Fact-Finding – Requirements Gathering – Evaluation and Rationalization – Prioritization – Integration and Validation.

### UNIT III REQUIREMENTS ANALYSIS 9

Identification of Functional and Non Functional Requirements – Identification of Performance Requirements – Identification of safety Requirements – Analysis – Feasibility and Internal Compatibility of System Requirements – Definition of Human Requirements Baseline.

### UNIT IV REQUIREMENTS DEVELOPMENT 9

Requirements analysis – Requirements Documentation – Requirements Development Workflow – Fundamentals of Requirements Development – Requirements Attributes Guidelines Document – Supplementary Specification Document – Use Case Specification Document – Methods for Software Prototyping – Evolutionary prototyping –Throwaway prototyping.



## UNIT V                      REQUIREMENTS VALIDATION

9

Validation objectives – Analysis of requirements validation – Activities – Properties – Requirement reviews – Requirements testing – Case tools for requirements engineering.

**TOTAL : 45 PERIODS**

### OUTCOMES

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

- Prepare SRS including the details of requirements engineering
- Describe the stages of requirements elicitation
- Analyze software requirements gathering

### REFERENCES:

1. Dean Leffingwe, Don Widrig, "Managing Software Requirements A Use Case Approach", Second Addition, Addison Wesley, 2003
2. Ian Graham, "Requirements Engineering and Rapid Development", Addison Wesley, 1998
3. Ian Sommerville, Pete Sawyer, "Requirements Engineering: A Good Practice Guide", Sixth Edition, Pearson Education, 2004
4. Karl Eugene Wiegers, "Software Requirements", Word Power Publishers, 2000
5. Wiegers, Karl, Joy Beatty, "Software requirements", Pearson Education, 2013

**CP5161**

## **DATA STRUCTURES LABORATORY**

**L T P C**  
**0 0 4 2**

### OBJECTIVES:

- To acquire the knowledge of using advanced tree structures.
- To learn the usage of heap structures.
- To understand the usage of graph structures and spanning trees.

### LIST OF EXPERIMENTS

Each student has to work individually on assigned lab exercises. Lab sessions could be scheduled as one contiguous four-hour session per week or two two-hour sessions per week. There will be about 15 exercises in a semester. It is recommended that all implementations are carried out in Java. If C or C++ has to be used, then the threads library will be required for concurrency. Exercises should be designed to cover the following topics:

### EXPERIMENTS:

1. Implementation of Merge Sort and Quick Sort-Analysis
2. Implementation of a Binary Search Tree
3. Red-Black Tree Implementation
4. Heap Implementation
5. Fibonacci Heap Implementation
6. Graph Traversals
7. Spanning Tree Implementation
8. Shortest Path Algorithms (Dijkstra's algorithm, Bellmann Ford Algorithm)
9. Implementation of Matrix Chain Multiplication
10. Activity Selection and Huffman Coding Implementation.

**TOTAL: 60 PERIODS**

**OUTCOMES:****Upon Completion of this course, the students should be able to:**

- Design and implement basic and advanced data structures extensively.
- Design algorithms using graph structures
- Design and develop efficient algorithms with minimum complexity using design techniques.

**CP5281****TERM PAPER WRITING AND SEMINAR****L T P C  
0 0 2 1**

In this course, students will develop their scientific and technical reading and writing skills that they need to understand and construct research articles. A term paper requires a student to obtain information from a variety of sources (i.e., Journals, dictionaries, reference books) and then place it in logically developed ideas. The work involves the following steps:

1. Selecting a subject, narrowing the subject into a topic
2. Stating an objective.
3. Collecting the relevant bibliography (atleast 15 journal papers)
4. Preparing a working outline.
5. Studying the papers and understanding the authors contributions and critically analysing each paper.
6. Preparing a working outline
7. Linking the papers and preparing a draft of the paper.
8. Preparing conclusions based on the reading of all the papers.
9. Writing the Final Paper and giving final Presentation

Please keep a file where the work carried out by you is maintained.  
Activities to be carried Out

<b>Activity</b>	<b>Instructions</b>	<b>Submission week</b>	<b>Evaluation</b>
Selection of area of interest and Topic	You are requested to select an area of interest, topic and state an objective	2 <sup>nd</sup> week	<b>3 %</b> Based on clarity of thought, current relevance and clarity in writing
Stating an Objective			
Collecting Information about your area & topic	<ol style="list-style-type: none"> <li>1. List 1 Special Interest Groups or professional society</li> <li>2. List 2 journals</li> <li>3. List 2 conferences, symposia or workshops</li> <li>4. List 1 thesis title</li> <li>5. List 3 web presences (mailing lists, forums, news sites)</li> <li>6. List 3 authors who publish regularly in your area</li> <li>7. Attach a call for papers</li> </ol>	3 <sup>rd</sup> week	<b>3%</b> ( the selected information must be area specific and of international and national standard)

	(CFP) from your area.		
Collection of Journal papers in the topic in the context of the objective – collect 20 & then filter	<ul style="list-style-type: none"> <li>You have to provide a complete list of references you will be using- Based on your objective -Search various digital libraries and Google Scholar</li> <li>When picking papers to read - try to: <ul style="list-style-type: none"> <li>Pick papers that are related to each other in some ways and/or that are in the same field so that you can write a meaningful survey out of them,</li> <li>Favour papers from well-known journals and conferences,</li> <li>Favour “first” or “foundational” papers in the field (as indicated in other people’s survey paper),</li> <li>Favour more recent papers,</li> <li>Pick a recent survey of the field so you can quickly gain an overview,</li> <li>Find relationships with respect to each other and to your topic area (classification scheme/categorization)</li> </ul> </li> <li>Mark in the hard copy of papers whether complete work or section/sections of the paper are being considered</li> </ul>	4 <sup>th</sup> week	<b>6%</b> ( the list of standard papers and reason for selection)
Reading and notes for first 5 papers	<p>Reading Paper Process</p> <ul style="list-style-type: none"> <li>For each paper form a Table answering the following questions: <ul style="list-style-type: none"> <li>What is the main topic of the article?</li> <li>What was/were the main issue(s) the author said they want to discuss?</li> </ul> </li> </ul>	5 <sup>th</sup> week	<b>8%</b> ( the table given should indicate your understanding of the paper and the evaluation is based on your conclusions about each paper)

	<ul style="list-style-type: none"> <li>• Why did the author claim it was important?</li> <li>• How does the work build on other's work, in the author's opinion?</li> <li>• What simplifying assumptions does the author claim to be making?</li> <li>• What did the author do?</li> <li>• How did the author claim they were going to evaluate their work and compare it to others?</li> <li>• What did the author say were the limitations of their research?</li> <li>• What did the author say were the important directions for future research?</li> </ul> <p>Conclude with limitations/issues not addressed by the paper (from the perspective of your survey)</p>		
Reading and notes for next 5 papers	Repeat Reading Paper Process	6 <sup>th</sup> week	<b>8%</b> (the table given should indicate your understanding of the paper and the evaluation is based on your conclusions about each paper)
Reading and notes for final 5 papers	Repeat Reading Paper Process	7 <sup>th</sup> week	<b>8%</b> (the table given should indicate your understanding of the paper and the evaluation is based on your conclusions about each paper)
Draft outline 1 and Linking papers	Prepare a draft Outline, your survey goals, along with a classification / categorization diagram	8 <sup>th</sup> week	<b>8%</b> (this component will be evaluated based on the linking and classification among the papers)
Abstract	Prepare a draft abstract and give a presentation	9 <sup>th</sup> week	<b>6%</b> (Clarity, purpose and conclusion) <b>6%</b> Presentation & Viva Voce

Introduction Background	Write an introduction and background sections	10 <sup>th</sup> week	5% ( clarity)
Sections of the paper	Write the sections of your paper based on the classification / categorization diagram in keeping with the goals of your survey	11 <sup>th</sup> week	10% (this component will be evaluated based on the linking and classification among the papers)
Your conclusions	Write your conclusions and future work	12 <sup>th</sup> week	5% ( conclusions – clarity and your ideas)
Final Draft	Complete the final draft of your paper	13 <sup>th</sup> week	10% (formatting, English, Clarity and linking) 4% Plagiarism Check Report
Seminar	A brief 15 slides on your paper	14 <sup>th</sup> & 15 <sup>th</sup> week	10% (based on presentation and Viva-voce)

**TOTAL : 30 PERIODS**

**SE5201**

**SOFTWARE TESTING METHODOLOGIES  
AND QUALITY ASSURANCE**

**L T P C  
3 0 2 4**

**OBJECTIVES**

**The student should be able to**

- Know what is software and the usage of different types of softwares.
- Know the Quality Metrics of various Softwares.
- Know the methodologies in making Software.
- Test the product finally to check the product Quality.

**UNIT I INTRODUCTION**

**9**

Introduction to Software Quality - Challenges – Objectives – Quality Factors – Components of SQA – Contract Review – Development and Quality Plans – SQA Components in Project Life Cycle – SQA Defect Removal Policies – Reviews.

**UNIT II TESTING METHODOLOGIES**

**9**

Basics of Software Testing – Test Generation from Requirements – Finite State Models – Combinatorial Designs - Test Selection, Minimization and Prioritization for Regression Testing – Test Adequacy, Assessment and Enhancement.

**UNIT III TEST STRATEGIES**

**9**

Testing Strategies – White Box and Black Box Approach – Integration Testing – System and Acceptance Testing – Performance Testing – Regression Testing - Internationalization Testing – Ad-hoc Testing – Website Testing – Usability Testing – Accessibility Testing.

**UNIT IV TEST AUTOMATION AND MANAGEMENT 9**  
Test plan – Management – Execution and Reporting – Software Test Automation – Automated Testing tools - Hierarchical Models of Software Quality – Configuration Management – Documentation Control.

**UNIT V SQA IN PROJECT MANAGEMENT 9**  
Project progress control – costs – quality management standards – project process standards – management and its role in SQA – SQA unit.

**TOTAL : 45 + 30 :75 PERIODS**

### **OUTCOMES**

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

- Analyze the product Quality.
- Use various testing methods.
- Assess Quality standards.

### **REFERENCES:**

1. Aditya Mathur, "Foundations of Software Testing", Pearson Education, 2008
2. Alan C Gillies, "Software Quality Theory and Management", Cengage Learning, Second Edition, 2003.
3. Daniel Galin, "Software Quality Assurance – from Theory to Implementation", Pearson Education, 2009
4. Robert Furtell, Donald Shafer, and Linda Shafer, "Quality Software Project Management", Pearson Education Asia, 2002.
5. Ron Patton, "Software Testing" , Second Edition, Pearson Education, 2007
6. Srinivasan Desikan, Gopalaswamy Ramesh, "Software Testing – Principles and Practices", Pearson Education, 2006
7. Yogesh Singh, "Software Testing", Cambridge University Press, 2012

**IF5251**

**SOFTWARE INDUSTRIALIZATION**

**L T P C  
3 0 0 3**

### **OBJECTIVES:**

**The student should be able:**

- To point out the need for industrialization in software development
- To understand the non functional requirements in software engineering
- To carry out performance analyses
- To study the various types of scalability
- To acquire the art of capacity planning
- To Understand the techniques for infrastructure management

**UNIT I INDUSTRIALIZATION OF SOFTWARE DEVELOPMENT 9**

The Fragile Hand Weaving – Features Vs Robustness – Components and Services Based Development – Agile and Dev Ops - Software Factory – Automation

**UNIT II NON FUNCTIONAL REQUIREMENTS and ENGINEERING 9**

NFRs - Cost of Quality – Business and System View – Industrialization Process in SDLC – Performance and Scalability – Capacity Planning –Production Operations

**UNIT III PERFORMANCE and SCALABILITY ENGINEERING****9**

Engineering for Performance and Scalability -Performance Modelling, Measurement and Testing – Workload Characterization – Latency and Throughput Requirements – Resource Usage Measurements Processor, Memory, Disk, Network – Performance Testing and Profiling – Bottleneck and Hotspot Identification – Vertical and Horizontal Scalability – Load, Space and Structural Scalability – Endurance Engineering – Analysis and Presenting Recommendations – Tools for Performance and Scalability

**UNIT IV THE ART OF CAPACITY PLANNING****9**

Capacity Planning Art Vs Science – Budgetary Capacity Planning - Utilization, Service Demand, The Forced Flow, Interactive Response Time, Little’s Laws – Using Queuing Models – Markov Models – M/M/1 M/G/1 Single Queue Systems – Mean Value Analysis- Multi Class Models – Priority Scheduling – Fork/Join Queuing Networks – Production Capacity Forecasting With Regression and Time Series Models – Tools for Capacity Planning

**UNIT V PRODUCTION SYSTEMS MANAGEMENT****9**

Infrastructure Management and Support – Systems, Storage and Network Monitoring – High Availability – Service Levels – Change and Configuration Management- Capacity Augmentation - Modernizing and Cloud Enablement - Automation

**TOTAL : 45 PERIODS****OUTCOMES:****At the end of this course, the students will be able to**

- Understand SOA and DevOps
- Understand the non-functional requirements in software engineering
- Apply various performance analysis techniques
- Analyze software systems for scalability
- Apply capacity planning methods
- Apply infrastructure management techniques

**REFERENCES:**

1. Andre B. Bondi, “Foundations of Software and System Performance Engineering”, Addison Wesley, 2015
2. Daniel A. Menasce, Dowdy, Almeida, “Computer Capacity Planning by Example”, Prentice Hall, 2004
3. L. Chung, B. Nixon, E. Yu and J. Mylopoulos, “Non-Functional Requirements in Software Engineering”, Springer, 2000
4. Rich Schiesser, “IT Systems Management”, Pearson Education, 2010

**OBJECTIVES****The student should be able to**

- Learn different definitions of software quality
- Know different notions of defects and classify them
- Understand the basic techniques of data collection and how to apply them
- Learn software metrics that define relevant metrics in a rigorous way.
- Gain confidence in ultra-high reliability.

**UNIT I INTRODUCTION TO SOFTWARE RELIABILITY 9**

Basic Concepts – Failure and Faults – Environment – Availability – Modeling – uses – requirements reliability metrics – design & code reliability metrics – testing reliability metrics.

**UNIT II SOFTWARE RELIABILITY MODELING 9**

Concepts – General Model Characteristic – Historical Development of models – Model Classification scheme – Markovian models – General concepts – General Poisson Type Models – Binomial Type Models – Poisson Type models – Fault reduction factor for Poisson Type models.

**UNIT III COMPARISON OF SOFTWARE RELIABILITY MODELS 9**

Comparison Criteria – Failure Data – Comparison of Predictive Validity of Model Groups – Recommended Models – Comparison of Time Domains – Calendar Time Modeling – Limiting Resource Concept – Resource Usage model – Resource Utilization – Calendar Time Estimation and confidence Intervals.

**UNIT IV FUNDAMENTALS OF MEASUREMENT 9**

Measurements in Software Engineering – Scope of Software metrics – Measurements theory – Goal based Framework – Software Measurement Validation.

**UNIT V MEASURING SOFTWARE PRODUCT 9**

Measurement of Internet Product Attributes – Size and Structure – External Product Attributes – Measurement of Quality – Software Reliability: Measurement and Prediction.

**TOTAL : 45 PERIODS****OUTCOMES:****At the end of this course, the students should be able to:**

- Perform some simple statistical analysis relevant to software measurement data.
- Use from practical examples both the benefits and limitations of software metrics for quality control and assurance

**REFERENCES:**

1. John D. Musa, "Software Reliability Engineering", Tata McGraw Hill, 1999
2. John D. Musa, Anthony Iannino, Kazuhira Okumoto, "Software Reliability – Measurement, Prediction, Application, Series in Software Engineering and Technology", McGraw Hill, 1987
3. Norman Fenton, James Bieman, "Software Metrics: A Rigorous and Practical Approach", 3<sup>rd</sup> edition, CRC Press, 2015



IF5191

**ADVANCED DATABASES**

<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

**OBJECTIVES:**

- To understand the design of databases.
- To acquire knowledge on parallel and distributed databases and its applications.
- To study the usage and applications of Object Oriented and Intelligent databases.
- To understand the emerging databases like Mobile, XML, Cloud and Big Data

**UNIT I                      PARALLEL AND DISTRIBUTED DATABASES                      9**

Database System Architectures: Centralized and Client-Server Architectures – Server System Architectures – Parallel Systems- Distributed Systems – Parallel Databases: I/O Parallelism – Inter and Intra Query Parallelism – Inter and Intra operation Parallelism – Design of Parallel Systems Distributed Database Concepts - Distributed Data Storage – Distributed Transactions – Commit Protocols – Concurrency Control – Distributed Query Processing – Case Studies

**UNIT II                      INTELLIGENT DATABASES                      9**

Active Databases: Syntax and Semantics (Starburst, Oracle, DB2)- Taxonomy- Applications- Design Principles for Active Rules- Temporal Databases: Overview of Temporal Databases TSQL2- Deductive Databases-Recursive Queries in SQL- Spatial Databases- Spatial Data Types - Spatial Relationships- Spatial Data Structures-Spatial Access Methods- Spatial DB Implementation.

**UNIT III                      XML DATABASES                      9**

XML Databases: XML Data Model – DTD – XML Schema – XML Querying – Web Databases – Open Database Connectivity.

**UNIT IV                      MOBILE DATABASES                      9**

Mobile Databases: Location and Handoff Management - Effect of Mobility on Data Management - Location Dependent Data Distribution - Mobile Transaction Models -Concurrency Control - Transaction Commit Protocols

**UNIT V                      MULTIMEDIA DATABASES                      9**

Multidimensional Data Structures – Image Databases – Text / Document Databases – Video Databases – Audio Databases – Multimedia Database Design.

**TOTAL :45 PERIODS**

**OUTCOMES:**

**Upon successful completion of this course, a student will be able to:**

- To develop skills on databases to optimize their performance in practice.
- To analyze each type of databases and its necessity
- To design faster algorithms in solving practical database problem

**REFERENCES:**

1. C.J.Date, A.Kannan, S.Swamynathan, “An Introduction to Database Systems”, Eighth Edition, Pearson Education, 2006.
2. Carlo Zaniolo, Stefano Ceri, Christos Faloutsos, Richard T.Snodgrass, V.S.Subrahmanian, Roberto Zicari, “Advanced Database Systems”, Morgan Kaufmann publishers,2006.
3. Henry F Korth, Abraham Silberschatz, S. Sudharshan, “Database System Concepts”, Sixth Edition, McGraw Hill, 2011.
4. R. Elmasri, S.B. Navathe, “Fundamentals of Database Systems”, Sixth Edition, Pearson Education/Addison Wesley, 2010.
5. Vijay Kumar, “Mobile Database Systems”, John Wiley & Sons, 2006.

**OBJECTIVES:**

To learn the stages of software development  
 To know about preparing software project documentation

**LIST of EXERCISES:****60**

**Choose any one application for performing the following phases.**

1. Program Analysis and Project Planning.  
 Thorough study of the problem – Identify project scope, Objectives, Infrastructure. –  
 PROJECT PLAN DOCUMENTATION
2. Software requirement Analysis  
 Describe the individual Phases / Modules of the project, Identify deliverables. – SRS  
 DOCUMENTATION
3. Data Modeling  
 Use work products – Data dictionary, Use case diagrams and activity diagrams, build  
 and test class diagrams, Sequence diagrams , add interface to class diagrams. –  
 DESIGN DOCUMENTATION
4. Software Development and Debugging  
 Use technology of your choice to develop and debug the application– CODE  
 DOCUMENTATION
5. Software Testing  
 Perform validation testing, Coverage analysis, memory leaks, develop test case  
 hierarchy, Site check and Site monitor. – TEST CASE DOCUMENTATION

**TOTAL : 60 PERIODS****SUGGESTED LIST OF APPLICATIONS:**

Student Marks Analyzing System.  
 Quiz System.  
 Online Ticket Reservation System  
 Payroll System  
 Course Registration System  
 Stock Maintenance.

**OUTCOMES:**

**At the end the student will be able to**

- Prepare project plan, SRS, design document, code document and test case documentations at appropriate stages of software development.

SE5301

**INTEGRATED SOFTWARE PROJECT  
MANAGEMENT**

**L T P C**  
**3 0 0 3**

**OBJECTIVES**

**The student should be able to**

- Understand the basic concept of project management.
- Learn the various costing and life cycle management.
- Understand the role played by risk in software project.
- Appreciate the use of metrics for software project management.
- Know the challenges in people management.

**UNIT I PROJECT MANAGEMENT & COSTING 9**

Software Project Management approaches – Project Acquisition – Initiation – Planning – PERT- Execution and Control – CPM – Change Management – Project Closure – Agile SPM Problems in Software Estimation – Algorithmic Cost Estimation Process, Function Points, COCOMO II (Constructive Cost Model) – Estimating Web Application Development – Concepts of Finance, Activity Based Costing and Economic Value Added (EVA) – Balanced Score Card.

**UNIT II PROCESS MODELS & LIFECYCLE MANAGEMENT 9**

Software Engineering Process Models - Adaptive Software Development (ASD) - DSDM - SCRUM – Crystal -Feature Driven Development (FDD) - ISO 9000: 2000 - SPICE – SIX SIGMA – CMMI. SLIM (Software Life cycle Management) – PLM (Product Lifecycle Management) – PDM (Product Data Management) - PLM, PDM Applications – Pre-PLM Environment – Change Management.

**UNIT III RISK MANAGEMENT 9**

Perspectives of Risk Management - Risk Definition – Risk Categories – Risk Assessment: Approaches, techniques and good practices – Risk Identification / Analysis / Prioritization – Risk Control (Planning / Resolution / Monitoring) – Risk Retention – Risk Transfer - Failure Mode and Effects Analysis (FMEA) – Operational Risks – Supply Chain Risk Management.

**UNIT IV METRICS 9**

Need for Software Metrics – scope – basics – framework for software measurement - Classification of Software Metrics: Product Metrics (Size Metrics, Complexity Metrics, Halstead's Product Metrics, Quality Metrics), and Process metrics (Empirical Models, Statistical Models, Theory-based Models, Composite Models, and Reliability Models) – measuring internal and external product attributes.

**UNIT V PEOPLE MANAGEMENT 9**

Leadership styles – Developing Leadership skills – Leadership assessment – Motivating People – Organizational strategy – Management – Team building – Delegation – Art of Interviewing People - Team Management – Rewarding - Client Relationship Management.

**TOTAL : 45 PERIODS**

**OUTCOMES**

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

- Identify the various elements of software management process framework
- Use available open source estimation tools for cost estimation
- Identify existing risk and perform risk assessment
- Design a software metric for software project management
- Modify the art of interviewing people for a given scenario.

## REFERENCES:

1. Antonio Borghesi, Barbara Gaudenzi, "Risk Management: How to Assess, Transfer and Communicate Critical Risks: Perspectives in Business Culture", Illustrated Edition, Springer, 2012
2. Murali Chemuturi, Thomas M. Cagley, "Mastering Software Project Management: Best Practices, Tools and Techniques", J. Ross Publishing, 2010
3. Norman Fenton, James Bieman, "Software Metrics: A Rigorous and Practical Approach", 3<sup>rd</sup> edition, CRC Press, 2015.
4. Stark, John, "Decision Engineering: Product Lifecycle Management: 21st Century Paradigm for Product Realisation", 2<sup>nd</sup> Edition., Springer London, 2011

**SE5001**

**AGILE SOFTWARE ENGINEERING**

L	T	P	C
3	0	0	3

## OBJECTIVES:

**The student should be able to**

- Understand agile software development practices
- Demonstrate Agile development and testing techniques
- Know the benefits and pitfalls of working in an Agile team
- Understand agile development and testing.

## UNIT I AGILE METHODOLOGY

**9**

Theories for Agile management – agile software development – traditional model vs. agile model - classification of agile methods – agile manifesto and principles – agile project management – agile team interactions – ethics in agile teams - agility in design, testing – agile documentations – agile drivers, capabilities and values.

## UNIT II AGILE PROCESSES

**9**

Lean production - SCRUM, Crystal, Feature Driven Development, Adaptive Software Development, and Extreme Programming: Method overview – lifecycle – work products, roles and practices.

## UNIT III AGILITY AND KNOWLEDGE MANAGEMENT

**9**

Agile information systems – agile decision making - Earl's schools of KM – institutional knowledge evolution cycle – development, acquisition, refinement, distribution, deployment , leveraging – KM in software engineering – managing software knowledge – challenges of migrating to agile methodologies – agile knowledge sharing – role of story-cards – Story-card Maturity Model (SMM).

## UNIT IV AGILITY AND REQUIREMENTS ENGINEERING

**9**

Impact of agile processes in RE – current agile practices – variance – overview of RE using agile – managing unstable requirements – requirements elicitation – agile requirements abstraction model – requirements management in agile environment, agile requirements prioritization – agile requirements modeling and generation – concurrency in agile requirements generation.

**UNIT V            AGILITY AND QUALITY ASSURANCE****9**

Agile Interaction Design - Agile product development – Agile Metrics – Feature Driven Development (FDD) – Financial and Production Metrics in FDD – Agile approach to Quality Assurance - Test Driven Development – Pair programming: Issues and Challenges - Agile approach to Global Software Development.

**TOTAL : 45 PERIODS****OUTCOMES:****At the end of this course, the students should be able to:**

- The know importance of interacting with business stakeholders in determining the requirements for a software system.
- Apply iterative software development process
- Apply the impact of social aspects on software development success.

**REFERENCES:**

1. Craig Larman, “Agile and Iterative Development: A manager’s Guide”, Addison-Wesley, 2004
2. David J. Anderson; Eli Schragenheim, “Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results”, Prentice Hall, 2003
3. Dingsoyr, Torgeir, Dyba, Tore, Moe, Nils Brede (Eds.), “Agile Software Development, Current Research and Future Directions”, Springer-Verlag Berlin Heidelberg, 2010
4. Hazza& Dubinsky, “Agile Software Engineering, Series: Undergraduate Topics in Computer Science”, Springer, VIII edition, 2009
5. Kevin C. Desouza, “Agile information systems: conceptualization, construction, and management”, Butterworth-Heinemann, 2007.

**SE5002****BUSINESS INTELLIGENCE**

<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

**OBJECTIVES:****The student should be able to**

- Identify the enormous opportunities that currently exists in providing business intelligence services
- Gain a practical understanding of the key data mining methods of classification, prediction, data reduction and exploration
- Understand and help develop the strategies of modern enterprise decision makers
- Acquire knowledge in many scientific and technological fields including data warehouses, data mining, content analytics, business process management, visual analytics
- Gain competences in information systems, web science, decision science, software engineering, and innovation and entrepreneurship.

**UNIT I            INTRODUCTION****9**

BI Basics – Meeting the BI challenge – BI user models – Basic reporting and querying – BI Markets - BI and Information Exploitation – Value of BI – BI cycle – Bridging the analysis gap – BI Technologies – BI Decision Support Initiatives – BI Project Team.





**REFERENCES:**

1. Danielle Ruest, Nelson Ruest, "Virtualization: A Beginner's Guide", McGraw-Hill Osborne Media, 2009.
2. Jim Smith, Ravi Nair , "Virtual Machines: Versatile Platforms for Systems and Processes", Elsevier/Morgan Kaufmann, 2005
3. John W.Rittinghouse and James F.Ransome, "Cloud Computing: Implementation, Management, and Security", CRC Press, 2010.
4. Kai Hwang, Geoffrey C Fox, Jack G Dongarra, "Distributed and Cloud Computing, From Parallel Processing to the Internet of Things", Morgan Kaufmann Publishers, 2012.
5. Tim Mather, Subra Kumaraswamy, and Shahed Latif , "Cloud Security and Privacy", O'Reilly Media, Inc.,2009.
6. Toby Velte, Anthony Velte, Robert Elsenpeter, "Cloud Computing, A Practical Approach", McGraw-Hill Osborne Media, 2009.
7. Tom White, "Hadoop: The Definitive Guide", Yahoo Press, 2012.

**SE5003**

**SOFTWARE TEST AUTOMATION**

<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

**OBJECTIVES**

**The student should be able to**

- Understand the basics of test automation
- Appreciate the different aspects of test tool evaluation and test automation approach selection
- Understand the role played by test planning and design in test execution
- Appreciate the use of various testing tools for testing varied applications
- Understand test automation using case studies

**UNIT I INTRODUCTION**

**9**

Fundamentals of test automation – Management issues – technical issues - Background on software testing – Automated test life cycle methodology (ATLM) – Test Maturity Model – Test Automation Development – Overcoming false expectations of automated testing – benefits – test tool proposal

**UNIT II TEST FRAMEWORK AND AUTOMATION**

**9**

Test Tool Evaluation and selection – organisations' system engineering environment – tools that support the testing life cycle – test process analysis – test tool consideration Test framework – Test Library Management –selecting the test automation approach - test team management

**UNIT III TEST PLANNING AND DESIGN**

**9**

Test planning – Test program scope – Test requirements management – Test Events, Activities and Documentation – Test Environment – Evolving a Test plan Test analysis and design – Test requirements analysis – Test program design – Test procedure design – Test development architecture – guidelines – automation infrastructure – test execution and review – test metrics

**UNIT IV TESTING THE APPLICATIONS**

**9**

Testing Web Applications – Functional Web testing with Twill – Selenium – Testing a simple Web Application – Testing Mobile Smartphone Applications – Running automated test scripts – Test tools for Browser based applications – Test Automation with Emulators



**UNIT V CASE STUDIES****9**

Test automation and agile project management – database automation – test automation in cloud – Mainframe and Framework automation – Model based test case generation – Model based testing of Android applications – exploratory test automation

**TOTAL : 45 PERIODS****OUTCOMES:**

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

- Identify the different test tools
- Use available testing tools to test some software applications
- Modify existing test metrics based on functionality or features used
- Design test cases and execute them
- Implement test scripts for automating test execution

**REFERENCES:**

1. C. Titus Brown, Gheorghe Gheorghiu, Jason Huggins, “An Introduction to Testing Web Applications with twill and Selenium”, O’Reilly Media, Inc., 2007
2. Dorothy Graham, Mark Fewster, “Experiences of Test Automation: Case Studies of Software Test Automation”, illustrated Edition, Addison-Wesley Professional, 2012
3. Elfriede Dustin, Jeff Rashka, “Automated software testing: Introduction, Management and Performance”, Pearson Education, 2008
4. Julian Harty, “A Practical Guide to Testing Mobile Smartphone Applications, Vol. 6 of Synthesis Lectures on Mobile and Pervasive Computing Series”, Morgan & Claypool Publishers, 2009
5. Kanglin Li, Mengqi Wu, “Effective Software Test Automation: Developing an Automated Software Testing Tool”, John Wiley & Sons, 2006
6. Linda Hayes, “The Automated Testing Handbook”, Software testing Inst., 1995
7. Mark Fewster, Dorothy Graham, “Software Test Automation”, Addison Wesley, 1999

**CP5293****BIG DATA ANALYTICS**

<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

**OBJECTIVES:**

- To understand the competitive advantages of big data analytics
- To understand the big data frameworks
- To learn data analysis methods
- To learn stream computing
- To gain knowledge on Hadoop related tools such as HBase, Cassandra, Pig, and Hive for big data analytics

**UNIT I INTRODUCTION TO BIG DATA****7**

Big Data – Definition, Characteristic Features – Big Data Applications - Big Data vs Traditional Data - Risks of Big Data - Structure of Big Data - Challenges of Conventional Systems - Web Data – Evolution of Analytic Scalability - Evolution of Analytic Processes, Tools and methods - Analysis vs Reporting - Modern Data Analytic Tools.

**UNIT II HADOOP FRAMEWORK 9**  
Distributed File Systems - Large-Scale FileSystem Organization – HDFS concepts - MapReduce Execution, Algorithms using MapReduce, Matrix-Vector Multiplication – Hadoop YARN

**UNIT III DATA ANALYSIS 13**  
Statistical Methods:Regression modelling, Multivariate Analysis - Classification: SVM & Kernel Methods - Rule Mining - Cluster Analysis, Types of Data in Cluster Analysis, Partitioning Methods, Hierarchical Methods, Density Based Methods, Grid Based Methods, Model Based Clustering Methods, Clustering High Dimensional Data - Predictive Analytics – Data analysis using R.

**UNIT IV MINING DATA STREAMS 7**  
Streams: Concepts – Stream Data Model and Architecture - Sampling data in a stream - Mining Data Streams and Mining Time-series data - Real Time Analytics Platform (RTAP) Applications - Case Studies - Real Time Sentiment Analysis, Stock Market Predictions.

**UNIT V BIG DATA FRAMEWORKS 9**  
Introduction to NoSQL – Aggregate Data Models – Hbase: Data Model and Implementations – Hbase Clients – Examples – .Cassandra: Data Model – Examples – Cassandra Clients – Hadoop Integration. Pig – Grunt – Pig Data Model – Pig Latin – developing and testing Pig Latin scripts. Hive – Data Types and File Formats – HiveQL Data Definition – HiveQL Data Manipulation – HiveQL Queries.

**TOTAL: 45 PERIODS**

**OUTCOMES:**

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

- Understand how to leverage the insights from big data analytics
- Analyze data by utilizing various statistical and data mining approaches
- Perform analytics on real-time streaming data
- Understand the various NoSql alternative database models

**REFERENCES:**

1. Bill Franks, "Taming the Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics", Wiley and SAS Business Series, 2012.
2. David Loshin, "Big Data Analytics: From Strategic Planning to Enterprise Integration with Tools, Techniques, NoSQL, and Graph", 2013.
3. Michael Berthold, David J. Hand, "Intelligent Data Analysis", Springer, Second Edition, 2007.
4. Michael Minelli, Michelle Chambers, and Ambiga Dhiraj, "Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses", Wiley, 2013.
5. P. J. Sadalage and M. Fowler, "NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence", Addison-Wesley Professional, 2012.
6. Richard Cotton ,Learning R – A Step-by-step Function Guide to Data Analysis, O'Reilly Media, 2013.

SE5004

KNOWLEDGE MANAGEMENT

L T P C  
3 0 0 3

**OBJECTIVES**

**The student should be able to**

- Learn knowledge engineering basics
- Know the knowledge models
- Know the techniques of knowledge management and implementation
- Learn the knowledge elicitation techniques
- Learn scope of knowledge management in project management.

**UNIT I INTRODUCTION 9**

The value of Knowledge – Knowledge Engineering Basics – Knowledge Economy – The Task and Organizational Content – Knowledge Management – Knowledge Management Ontology.

**UNIT II KNOWLEDGE MODELS 9**

Knowledge Model Components – Template Knowledge Models – Reflective Knowledge Models – Knowledge Model Construction – Types of Knowledge Models.

**UNIT III TECHNIQUES OF KNOWLEDGE MANAGEMENT 9**

Knowledge Elicitation Techniques – Modeling Communication Aspects – Knowledge Management and Organizational Learning.

**UNIT IV KNOWLEDGE SYSTEM IMPLEMENTATION 9**

Case Studies – Designing Knowledge Systems – Knowledge Codification – Testing and Deployment – Knowledge Transfer and Knowledge Sharing – Knowledge System Implementation.

**UNIT V ADVANCED KNOWLEDGE MANAGEMENT 9**

Advanced Knowledge Modeling – Value Networks – Business Models for Knowledge Economy – UML Notations – Project Management.

**TOTAL : 45 PERIODS**

**OUTCOMES:**

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

- Apply knowledge engineering basics.
- Design the knowledge models.
- Apply the techniques of knowledge management and implementation.

**REFERENCES:**

1. Awad, Elias M., and Hassan M. Ghaziri. "Knowledge Management", Prentice Hall; United States ed edition, 2011
2. C.W. Holsapple, "Handbooks on Knowledge ManagementII", International Handbooks on Information Systems, Vol 1 and 2, 2003.
3. Debowski, Shelda, "Knowledge Management: A Strategic Management Perspective", John Wiley & Sons Ltd, 2005
4. Elias M.Awad& Hassan M.Ghaziri, "Knowledge Management", Pearson Education, 2003
5. Guus Schreiber, Hans Akkermans, Anjo Anjewierden, Robert de Hoog, Nigel Shadbolt, Walter Van de Velde and Bob Wielinga, "Knowledge Engineering and Management", Universities Press, 2001

SE5005

**SOFTWARE VERIFICATION AND VALIDATION**

<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

**OBJECTIVES**

**The student should be able to**

- Understand the principles of verification and validation
- Appreciate the different verification and validation techniques
- Understand the various stages of testing
- Appreciate the use of tools for verification and validation
- Appreciate the benefits of using metrics for verification and validation

**UNIT I INTRODUCTION**

**9**

Principles of verification and validation – software architecture frameworks – model driven architecture – UML – systems modeling language – verification, validation and accreditation.

**UNIT II METHODS OF SOFTWARE VERIFICATION**

**9**

Verification and validation life cycle – traceability analysis – interface analysis – design and code verification – test analysis - Reviews – inspections - walkthroughs – audits – tracing – formal proofs – Model based verification and validation - Program verification techniques – formal methods of software verification – clean room methods.

**UNIT III TESTING**

**9**

Stages of Testing: Test Planning – Test design – Test case definition – Test procedure – Test reporting – Unit testing: white box , black box and performance testing – system testing: Function, performance, interface, operations, resource, security, portability, reliability, maintainability, safety, regression and stress testing – integration testing – acceptance testing: capability, constraint testing - structured testing – structured integration testing

**UNIT IV TOOLS FOR SOFTWARE VERIFICATION**

**9**

Tools for verification and validation: static analyser – configuration management tools – reverse engineering tools – tracing tools – tools for formal analysis – tools for testing – test case generators – test harnesses – debuggers – coverage analysers – performance analysers – test management tools

**UNIT V ADVANCED APPROACHES**

**9**

Automatic approach for verification and validation – validating UML behavioral diagrams – probabilistic model checking of activity diagrams in SysML – metrics for verification and validation

**TOTAL : 45 PERIODS**

**OUTCOMES:**

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

- Identify the different techniques for verification and validation
- Use available traceability analysis tools on sample requirements
- Modify existing coverage analysers in terms of functionality or features used
- Design system test cases
- Use test case generators and test management tools

## REFERENCES:

1. Avner Engel, "Verification, Validation & Testing of Engineered Systems", Wiley series in systems Engineering and Management, 2010.
2. ESA Board for Software Standardisation and Control (BSSC), "Guide to software verification and Validation", European Space Agency ESA PSS-05-10 Issue 1 Revision 1, 1995
3. Marcus S. Fisher, "Software Verification and Validation: An Engineering and Scientific Approach", Springer, 2007
4. Mourad Debbabi, Hassaine F, Jarrya Y., Soeanu A., Alawneh L., "Verification and Validation in Systems Engineering", Springer, 2010

CP5191

MACHINE LEARNING TECHNIQUES

L T P C  
3 0 0 3

## OBJECTIVES:

- To introduce students to the basic concepts and techniques of Machine Learning.
- To have a thorough understanding of the Supervised and Unsupervised learning techniques
- To study the various probability based learning techniques
- To understand graphical models of machine learning algorithms

### UNIT I INTRODUCTION

9

Learning – Types of Machine Learning – Supervised Learning – The Brain and the Neuron – Design a Learning System – Perspectives and Issues in Machine Learning – Concept Learning Task – Concept Learning as Search – Finding a Maximally Specific Hypothesis – Version Spaces and the Candidate Elimination Algorithm – Linear Discriminants – Perceptron – Linear Separability – Linear Regression.

### UNIT II LINEAR MODELS

9

Multi-layer Perceptron – Going Forwards – Going Backwards: Back Propagation Error – Multi-layer Perceptron in Practice – Examples of using the MLP – Overview – Deriving Back-Propagation – Radial Basis Functions and Splines – Concepts – RBF Network – Curse of Dimensionality – Interpolations and Basis Functions – Support Vector Machines

### UNIT III TREE AND PROBABILISTIC MODELS

9

Learning with Trees – Decision Trees – Constructing Decision Trees – Classification and Regression Trees – Ensemble Learning – Boosting – Bagging – Different ways to Combine Classifiers – Probability and Learning – Data into Probabilities – Basic Statistics – Gaussian Mixture Models – Nearest Neighbor Methods – Unsupervised Learning – K means Algorithms – Vector Quantization – Self Organizing Feature Map

### UNIT IV DIMENSIONALITY REDUCTION AND EVOLUTIONARY MODELS

9

Dimensionality Reduction – Linear Discriminant Analysis – Principal Component Analysis – Factor Analysis – Independent Component Analysis – Locally Linear Embedding – Isomap – Least Squares Optimization – Evolutionary Learning – Genetic algorithms – Genetic Offspring: - Genetic Operators – Using Genetic Algorithms – Reinforcement Learning – Overview – Getting Lost Example – Markov Decision Process

## UNIT V GRAPHICAL MODELS

9

Markov Chain Monte Carlo Methods – Sampling – Proposal Distribution – Markov Chain Monte Carlo – Graphical Models – Bayesian Networks – Markov Random Fields – Hidden Markov Models – Tracking Methods

**TOTAL: 45 PERIODS**

### OUTCOMES:

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

- Distinguish between, supervised, unsupervised and semi-supervised learning
- Apply the apt machine learning strategy for any given problem
- Suggest supervised, unsupervised or semi-supervised learning algorithms for any given problem
- Design systems that uses the appropriate graph models of machine learning
- Modify existing machine learning algorithms to improve classification efficiency

### REFERENCES:

1. Ethem Alpaydin, “Introduction to Machine Learning 3e (Adaptive Computation and Machine Learning Series)”, Third Edition, MIT Press, 2014
2. Jason Bell, “Machine learning – Hands on for Developers and Technical Professionals”, First Edition, Wiley, 2014
3. Peter Flach, “Machine Learning: The Art and Science of Algorithms that Make Sense of Data”, First Edition, Cambridge University Press, 2012.
4. Stephen Marsland, “Machine Learning – An Algorithmic Perspective”, Second Edition, Chapman and Hall/CRC Machine Learning and Pattern Recognition Series, 2014.
5. Tom M Mitchell, “Machine Learning”, First Edition, McGraw Hill Education, 2013.

**SE5006**

## **VIRTUALIZATION TECHNIQUES AND APPLICATIONS**

L	T	P	C
3	0	0	3

### OBJECTIVES

- To understand the concepts of virtualization and virtual machines
- To understand the implementation of process and system virtual machines
- To explore the aspects of high level language virtual machines
- To gain expertise in server, network and storage virtualization.
- To understand and deploy practical virtualization solutions and enterprise solutions

## UNIT I OVERVIEW OF VIRTUALIZATION

9

System architectures - Virtual Machine basics - Process vs System Virtual Machines - Taxonomy. Emulation: Basic Interpretation - Threaded Interpretation - Pre-coded and Direct Threaded Interpretation - Binary Translation. System Virtual Machines - Key concepts - Resource utilization basics

## UNIT II PROCESS VIRTUAL MACHINES

9

Implementation – Compatibility – Levels – Framework – State Mapping – Register – Memory Address Space – Memory Architecture Emulation – Memory Protection – Instruction Emulation – Performance Tradeoff - Staged Emulation – Exception Emulation – Exception Detection – Interrupt Handling – Operating Systems Emulation – Same OS Emulation – Different OS Emulation – System Environment

## UNIT III HIGH LEVEL LANGUAGE VIRTUAL MACHINES AND SERVER VIRTUALIZATION

9

HLL virtual machines: Pascal P-Code – Object Oriented HLLVMs - Java VM architecture - Java Native Interface - Common Language Infrastructure. Server virtualization: Partitioning techniques - virtual hardware - uses of virtual servers - server virtualization platforms

**UNIT IV NETWORK AND STORAGE VIRTUALIZATION 9**

Design of Scalable Enterprise Networks – Layer2 Virtualization – VLAN - VFI - Layer 3 Virtualization – VRF - Virtual Firewall Contexts - Network Device Virtualization - Data- Path Virtualization - Routing Protocols. Hardware Devices – SAN backup and recovery techniques – RAID – Classical Storage Model – SNIA Shared Storage Model – Virtual Storage: File System Level and Block Level

**UNIT V APPLYING VIRTUALIZATION 9**

Practical Virtualization Solutions: Comparison of Virtualization Technologies: Guest OS/ Host OS – Hypervisor – Emulation – Kernel Level – Shared Kernel, Enterprise Solutions: VMWare Server – VMWare ESXi – Citrix Xen Server – Microsoft Virtual PC – Microsoft Hyper-V – Virtual Box, Server Virtualization: Configuring Servers with Virtualization – Adjusting and Tuning Virtual servers – VM Backup – VM Migration, Desktop Virtualization: Terminal services – Hosted Desktop – Web-based Solutions – Localized Virtual Desktops, Network and Storage Virtualization: Virtual Private Networks – Virtual LAN – SAN and VSAN – NAS

**TOTAL : 45 PERIODS**

**OUTCOMES:**

**Upon completion of this course, the student should be able to**

- Deploy legacy OS on virtual machines.
- Analyze the intricacies of server, storage and network virtualizations
- Design and develop applications on virtual machine platforms

**REFERENCES:**

1. Chris Wolf, Erick M. Halter, “Virtualization: From the Desktop to the Enterprise”, APress 2005.
2. David Marshall, Wade A. Reynolds, “Advanced Server Virtualization: VMware and Microsoft Platform in the Virtual Data Center”, Auerbach Publications, 2006.
3. James E. Smith, Ravi Nair, “Virtual Machines: Versatile Platforms for Systems and Processes”, Elsevier/Morgan Kaufmann, 2005.
4. Kenneth Hess , Amy Newman, “Practical Virtualization Solutions: Virtualization from the Trenches”, Prentice Hall, 2010
5. Kumar Reddy, Victor Moreno, “Network virtualization”, Cisco Press, July, 2006.

<b>CP5292</b>	<b>INTERNET OF THINGS</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

**OBJECTIVES:**

- To understand the fundamentals of Internet of Things
- To learn about the basics of IOT protocols
- To build a small low cost embedded system using Raspberry Pi.
- To apply the concept of Internet of Things in the real world scenario.

**UNIT I INTRODUCTION TO IoT 9**

Internet of Things - Physical Design- Logical Design- IoT Enabling Technologies - IoT Levels & Deployment Templates - Domain Specific IoTs - IoT and M2M - IoT System Management with NETCONF-YANG- IoT Platforms Design Methodology

**UNIT II IoT ARCHITECTURE 9**

M2M high-level ETSI architecture - IETF architecture for IoT - OGC architecture - IoT reference model - Domain model - information model - functional model - communication model - IoT reference architecture

**UNIT III IoT PROTOCOLS 9**  
 Protocol Standardization for IoT – Efforts – M2M and WSN Protocols – SCADA and RFID Protocols – Unified Data Standards – Protocols – IEEE 802.15.4 – BACNet Protocol – Modbus– Zigbee Architecture – Network layer – 6LowPAN - CoAP - Security

**UNIT IV BUILDING IoT WITH RASPBERRY PI & ARDUINO 9**  
 Building IOT with RASPBERRY PI- IoT Systems - Logical Design using Python – IoT Physical Devices & Endpoints - IoT Device -Building blocks -Raspberrry Pi -Board - Linux on Raspberrry Pi - Raspberrry Pi Interfaces -Programming Raspberrry Pi with Python - Other IoT Platforms - Arduino.

**UNIT V CASE STUDIES AND REAL-WORLD APPLICATIONS 9**  
 Real world design constraints - Applications - Asset management, Industrial automation, smart grid, Commercial building automation, Smart cities - participatory sensing - Data Analytics for IoT – Software & Management Tools for IoT Cloud Storage Models & Communication APIs - Cloud for IoT - Amazon Web Services for IoT.

**TOTAL : 45 PERIODS**

**OUTCOMES:**

**Upon completion of the course, the student should be able to:**

- Analyze various protocols for IoT
- Develop web services to access/control IoT devices.
- Design a portable IoT using Rasperry Pi
- Deploy an IoT application and connect to the cloud.
- Analyze applications of IoT in real time scenario

**REFERENCES:**

1. Arshdeep Bahga, Vijay Madiseti, “Internet of Things – A hands-on approach”, Universities Press, 2015
2. Dieter Uckelmann, Mark Harrison, Michahelles, Florian (Eds), “Architecting the Internet of Things”, Springer, 2011.
3. Honbo Zhou, “The Internet of Things in the Cloud: A Middleware Perspective”, CRC Press, 2012.
4. Jan Ho” ller, Vlasios Tsiatsis , Catherine Mulligan, Stamatias , Karnouskos, Stefan Avesand. David Boyle, "From Machine-to-Machine to the Internet of Things - Introduction to a New Age of Intelligence", Elsevier, 2014.
5. Olivier Hersent, David Boswarthick, Omar Elloumi , “The Internet of Things – Key applications and Protocols”, Wiley, 2012

<b>SE5007</b>	<b>WEB DESIGN AND MANAGEMENT</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

**OBJECTIVES**

The student should be able to

- Know the importance of web technologies for the real world applications
- Learn appropriate scripting languages
- Know the testing techniques to test the product
- Gain the skills and project-based experience needed for entry into web design and development careers.
- To use a variety of strategies and tools to create websites.



**UNIT I SITE ORGANIZATION AND NAVIGATION 9**

User centered design – Web medium – Web design process – Evaluating process – Site types and architectures – Navigation theory – Basic navigation practices – Search – Site maps.

**UNIT II ELEMENTS OF PAGE DESIGN 9**

Browser compatible design issues - Pages and Layout – Templates – Text – Color – Images – Graphics and Multimedia - GUI Widgets and Forms – Web Design patterns

**UNIT III SCRIPTING LANGUAGES 9**

Client side scripting: XHTML – DHTML– JavaScript– XML Server side scripting: Perl – PHP – ASP/JSP Designing a Simple web application

**UNIT IV PRE-PRODUCTION MANAGEMENT 9**

Principles of Project Management – Web Project Method – Project Road Map – Project Clarification – Solution Definition – Project Specification – Content – Writing and Managing content.

**UNIT V PRODUCTION, MAINTENANCE AND EVALUATION 9**

Design and Construction – Testing, Launch and Handover – Maintenance – Review and Evaluation – Case Study.

**TOTAL : 45 PERIODS**

**OUTCOMES:**

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

- Apply various scripting languages for the development of web applications
- Follow Web design standards
- Develop websites for local community organizations.

**REFERENCES:**

1. Ashley Friedlein, "Web Project Managementll", Morgan Kaufmann Publishers, 2001
2. H. M. Deitel, P. J. Deitel, A. B. Goldberg, "Internet and World Wide Web – How to Program", Third Edition, Pearson Education 2004
3. Joel Sklar, "Principles of Web Design", Thomson Learning, 2001
4. Lynch, Horton and Rosenfeld, "Web Style Guide: Basic Design Principles for Creating Web Sites", 2nd Edition, Yale University Press, 2002
5. Thomas A. Powell, "The Complete Reference – Web Design", Tata McGraw Hill, Third Edition, 2003
6. Van Duyne, Landay, Hong, "The Design of Sites: Patterns for creating winning web sites", 2nd Edition, Prentice Hall, 2006
7. Wendy Willard, "Web Design: A Beginner's Guide", Second Edition, McGraw Hill Education (India) Private Limited, 2010

**OBJECTIVES**

- To gain knowledge about the current web development and emergence of social web
- To study about the modeling, aggregating and knowledge representation of semantic web
- To appreciate the use of machine learning approaches for web content mining
- To learn about the extraction and mining tools for social networks
- To gain knowledge on web personalization and web visualization of social networks

**UNIT I INTRODUCTION TO SOCIAL NETWORK ANALYSIS AND  
KNOWLEDGE REPRESENTATION 9**

Introduction to Web - Limitations of current Web – Development of Semantic Web – Emergence of the Social Web - Network analysis - Development of Social Network Analysis - Key concepts and measures in network analysis- Knowledge Representation on the Semantic Web – Ontology languages for the Semantic Web – RDF and OWL - Modeling and aggregating social network data.

**UNIT II SOCIAL MEDIA MINING 9**

Data Mining Essential –Data Mining Algorithm - Web Content Mining – Supervised Learning – Decision tree - Naïve Bayesian Text Classification - Support Vector Machines - Ensemble of Classifiers. Unsupervised Learning - K-means Clustering - Hierarchical Clustering –Partially Supervised Learning – Markov Models - Probability-Based Clustering - Classification and Clustering – Vector Space Model – Latent semantic Indexing – Automatic Topic Extraction - Opinion Mining and Sentiment Analysis – Document Sentiment Classification

**UNIT III EXTRACTION AND MINING COMMUNITITES IN WEB SOCIAL  
NETWORKS 9**

Extracting evolution of Web Community from a Series of Web Archive - Detecting Communities in Social Networks - Definition of Community - Evaluating Communities - Methods for Community Detection & Mining - Applications of Community Mining Algorithms - Tools for Detecting Communities Social Network Infrastructures and Communities - Decentralized Online Social Networks- Multi-Relational Characterization of Dynamic Social Network Communities

**UNIT IV HUMAN BEHAVIOR ANALYSIS AND PRIVACY ISSUES 9**

Understanding and Predicting Human Behavior for Social Communities - User Data Management, Inference and Distribution - Enabling New Human Experiences - Reality Mining - Context-Awareness - Privacy in Online Social Networks - Trust in Online Environment - Trust Models Based on Subjective Logic - Trust Network Analysis - Trust Transitivity Analysis - Combining Trust and Reputation – Trust Derivation Based on Trust Comparisons - Attack Spectrum and Countermeasures.

**UNIT V VISUALIZATION AND APPLICATIONS OF SOCIAL  
NETWORKS 9**

Graph Theory- Centrality- Clustering - Node-Edge Diagrams, Matrix representation, Visualizing Online Social Networks, Visualizing Social Networks with Matrix-Based Representations- Matrix +Node-Link Diagrams, Hybrid Representations - Applications - Covert Networks - Community Welfare - Collaboration Networks - Co-Citation Networks- Recommendation in Social Media: Challenges- Classical Recommendation Algorithms-Recommendation Using Social Context-Evaluating Recommendations.

**TOTAL : 45 PERIODS**

**OUTCOMES:**

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

- Apply knowledge for current Web development in the era of social Web
- Model, aggregate and represent knowledge for Semantic Web
- Use machine learning approaches for Web Content Mining
- Design extraction and mining tools for Social networks
- Develop personalized web sites and visualization for Social networks

**REFERENCES:**

1. Bing Liu, "Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data (Data-Centric Systems and Applications)", Springer; Second Edition, 2011.
2. Borko Furht, "Handbook of Social Network Technologies and Applications", Springer, 2010.
3. Dion Goh and Schubert Foo, "Social information retrieval systems: emerging technologies and Applications for searching the Web effectively", Idea Group, 2007.
4. Guandong Xu, Yanchun Zhang and Lin Li, "Web Mining and Social Networking Techniques and applications", Springer, 2011.
5. John G. Breslin, Alexandre Passant and Stefan Decker, "The Social Semantic Web", Springer, 2010.
6. Max Chevalier, Christine Julien and Chantal Soulé-Dupuy, "Collaborative and social Information retrieval and access: Techniques for Improved User Modelling", Information Science Reference, 2009.
7. Peter Mika, "Social networks and the Semantic Web", Springer, 2007.
8. Reza Zafarani, Mohammad Ali Abbasi, Huan Liu, "Social Media Mining", Cambridge University Press, 2014.

**SE5009**

**TEST DRIVEN DEVELOPMENT**

<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

**OBJECTIVES**

**The student should be able to**

- Understand TDD and MDD
- Learn fundamentals of refactoring
- Perform unit testing in TDD
- Apply TDD
- Appreciate the emphasis of TDD in extreme programming.

**UNIT I BACKGROUND**

**9**

Software Development – Evolution – Agile environments - Test Driven Development – definition – importance - TDD and traditional testing – TDD vs. MDD, TDD vs. Acceptance TDD – Fakes , Mocks and Integration tests - TDD and documentation - Test-driven database development - Scaling TDD via Agile Model-Driven Development - Shortcomings – TDD Tools

**UNIT II REFACTORING**

**9**

Principles in Refactoring - Building Tests - Toward a Catalog of Refactorings - Composing Methods - Moving Features Between Objects - Organizing Data - Simplifying Conditional Expressions - Dealing with Generalization - Big Refactorings - Refactoring, Reuse, and Reality - Refactoring Tools.

**UNIT III UNIT TESTING****9**

Unit Testing – Motivators – Types of Tests – Improving Assertion – Improving Test Cases – Improving Test Suites – JUnit – Fundamentals – Developing an application with JUnit.

**UNIT IV PATTERNS FOR TEST DRIVEN DEVELOPMENT****9**

Test Driven Development Patterns – Red Bar Patterns – Testing Patterns – Green Bar Patterns – xUnit Patterns – Design Patterns – Mastering TDD.

**UNIT V EXTREME PROGRAMMING****9**

Extreme Principles – Creating a Vision of the System – Writing Acceptance Testing – Planning – Providing Estimates – Planning Releases – Planning Iterations – Tactical Planning – Pair Programming – Test First – Design – Code with intension – Relentless Integration – Delivering the System.

**TOTAL : 45 PERIODS****OUTCOMES:****At the end the student will be able to**

- Differentiate Test Driven Development and Model Driven Development
- Apply Refactoring for improving code
- Use Test Driven Development patterns in project development.

**REFERENCES:**

1. Dave Astels, "Test driven development: A practical guide", Prentice Hall Professional Technical Reference, 2003
2. David Astels, Granville Miller, Miroslav Novak, "The Practical Guide to Extreme Programming", Prentice Hall PTR, 2002
3. Fowler, Martin, "Refactoring: improving the design of existing code", Pearson Education India, 2002
4. Kent Beck, "Test-driven development: by example", Addison-Wesley Professional, 2003
5. Lasse Koskela. Test Driven. Manning, 2007
6. Lech Madeyski, "Test-driven development: An empirical evaluation of agile practice", Springer Science & Business Media, 2009
7. Roy Oshero. "The art of unit testing: with examples in." NET 2009

**SE5010****PERSONAL SOFTWARE PROCESS**

L	T	P	C
3	0	0	3

**OBJECTIVES****The student should be able to**

- Understand the nature of PSP
- Apply PSP principles in measuring software
- Appreciate the role of PSP in assessing software quality
- Relate PSP and TSP in software development.
- Learn to use PSP in Software engineering.

**UNIT I INTRODUCTION 9**

Personal Process Strategy – PSP Purpose – Logic for Software Engineering Discipline – Operational Processes – Defining and Using a Personal Process – Learning to Use a Personal Process – Baseline Personal Process – Contents – PSP Process Elements – PSP Structure and Levels – Incremental Development – PSP Tool Support.

**UNIT II PSP SIZE ESTIMATION 9**

Measuring Software Size – Size Measures – Establishing a Database Counting Standard - Establishing a LOC Counting Standard – Size Accounting – Using Size Data – Calculating Productivity – Size Counters – Other Size Measures – Software Estimating – Principles – Conceptual Design – Proxy Based Estimating – Producing Relative Size Table – Estimating Considerations – Probe Estimating Method.

**UNIT III PSP QUALITY MANAGEMENT 9**

PSP Quality Strategy – Software Quality – Economics of Software Quality – Defect Types – Personal Quality Practices – Quality Measures – Quality Management – Managing Product Quality – PSP Improvement Practices – Defect Prevention.

**UNIT IV PSP DESIGN TEMPLATE 9**

Design Process – Design Levels – Design Strategies – Design Quality – Design Representation – Design Templates (Operational, Functional, State and Logic) – State machine design example – Using PSP Design in Large Scale Design – Design Verification.

**UNIT V TEAM SOFTWARE PROCESS 9**

Development – Structure of TSP – Launching TSP Team – TSP Team working Process – TSP Quality Management – TSP Status and Future Trends.

**TOTAL : 45 PERIODS**

**OUTCOMES**

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

- Analyse software using PSP
- Use PSP tools to measure software quality
- Use PSP in software design

**REFERENCES:**

1. Marsha Pomeroy-Huff, Robert Cannon, Timothy A. Chick, Julia Mullaney, and William Nichols, "The Personal Software Process SM (PSP SM) Body of Knowledge, Version 2.0", 2009
2. Watts S Humphrey, "PSP (SM): a self-improvement process for software engineers", Addison-Wesley Professional, 2005
3. Watts S Humphrey, "Team Software Process (TSP)", John Wiley & Sons, Inc., 2000.

**OBJECTIVES****The student should be able to**

- Know the importance and need of software security
- Know about various attacks
- Learn about secure software design
- Understand risk management in secure software development
- Know the working of tools related to software security

**UNIT I INTRODUCTION****9**

Need for software security – Memory based attacks – low level attacks against heap and stack -stack smashing – format string attacks – stale memory access attacks – ROP (Return oriented programming) – malicious computation without code injection. Defense against memory based attacks – stack canaries – non-executable data - address space layout randomization (ASLR), memory-safety enforcement, control-flow Integrity (CFI) – randomization

**UNIT II SECURE DESIGN****9**

Isolating the effects of untrusted executable content - stack inspection – policy specification languages – vulnerability trends – buffer overflow – code injection - Generic network fault injection – local fault injection - SQL injection - Session hijacking. Secure design - threat modeling and security design principles - good and bad software design - Web security-browser security: cross-site scripting (XSS) , cross-site forgery (CSRF) – database security – file security.

**UNIT III SECURITY RISK MANAGEMENT****9**

Risk Management Life cycle – Risk Profiling – Risk exposure factors – Risk Evaluation and Mitigation - Risk Assessment Techniques – Threat and Vulnerability Management.

**UNIT IV SECURITY TESTING****9**

Traditional software testing – comparison - secure software development life cycle - risk based security testing – prioritizing security testing with threat modeling – shades of analysis: white, grey and black box testing.

**UNIT V ADVANCED SOFTWARE SECURITY****9**

Advanced penetration testing – planning and scoping – DNS groper – DIG (Domain Information Graph) – Enumeration – Remote Exploitation – Web Application Exploitation - Exploits and Client side Attacks – Post Exploitation – Bypassing Firewalls and Avoiding Detection - Tools for penetration testing

**TOTAL : 45 PERIODS****OUTCOMES:****At the end the student will be able to**

- Use tools for securing software
- Apply security principles in software development
- Involve selection of testing techniques related to software security in testing phase of software development

## REFERENCES:

1. Bryan Sullivan and Vincent Liu, "Web Application Security, A Beginner's Guide", Kindle Edition, McGraw Hill, 2012
2. Chris Wysopal, Lucas Nelson, Dino Dai Zovi, and Elfriede Dustin, "The Art of Software Security Testing: Identifying Software Security Flaws (Symantec Press)", Addison-Wesley Professional, 2006
3. Evan Wheeler, "Security Risk Management: Building an Information Security Risk Management Program from the Ground Up", First edition, Syngress Publishing, 2011
4. Lee Allen, "Advanced Penetration Testing for Highly-Secured Environments: The Ultimate Security Guide (Open Source: Community Experience Distilled)", Kindle Edition, Packt Publishing, 2012
5. Mike Shema, "Hacking Web Apps: Detecting and Preventing Web Application Security Problems", First edition, Syngress Publishing, 2012
6. Robert C. Seacord, "Secure Coding in C and C++ (SEI Series in Software Engineering)", Addison-Wesley Professional, 2005

<b>SE5012</b>	<b>ENTERPRISE APPLICATION INTEGRATION</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

## OBJECTIVES

### The student should be able to

- Create a Business Data Catalog Application Definition file
- Discuss the most important topics and technologies related to enterprise application integration
- Practice implementation strategies.
- Know the Integration facilities for an application.
- To appreciate the current trends in Enterprise Application Integration.

## **UNIT I INTRODUCTION 9**

Business Imperative for Enterprise Integration – Business agility – Business transactions – need of E-business - ROI of Enterprise Integration – Challenges – Business drivers – Defining Requirements – Enterprise Integration strategy.

## **UNIT II ENTERPRISE INTEGRATION ARCHITECTURE 9**

Overview – Business case – Components of EIA – Organizational Structure – Architectural Governance - Understanding Integration Technology – Current Integration Architecture – Technical Integration Architecture specification.

## **UNIT III SERVICE AND INFORMATION INTEGRATION ARCHITECTURE 9**

Service Oriented Architecture – Benefits – Defining Services – Event driven service design – specification – Understanding Metadata – Metadata Architecture – standards – Information Integration Patterns – Architecture Specification.

## **UNIT IV PROCESS AND APPLICATION INTEGRATION ARCHITECTURE 9**

Process to Business – Process Integration Technology – Process Standards – Architecture Specification - Choosing Technology - Application Integration Technology – Implementation Specification – Composite Application – Composite integration specification.

**UNIT V                    PATTERNS OF ENTERPRISE APPLICATION INTEGRATION                    9**

Domain Logic Patterns – Data source architectural patterns – Object Relational Patterns – Web Presentation Patterns – Distribution Patterns – Concurrency Patterns – Session state patterns - base patterns.

**TOTAL : 45 PERIODS**

**OUTCOMES:**

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

- Use object-oriented concepts for developing web applications
- Demonstrate the enterprise application integration
- Deploy the components of EIA with solutions

**REFERENCES:**

1. David S.Linthicum, —Enterprise Application Integration, Addison-Wesley Professional, 2000
2. Gregor Hohpe, Bobby Woolf, “Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions”, Pearson Education, 15<sup>th</sup> edition, 2011.
3. Martin Fowler, “Patterns of Enterprise Application Architecture”, Pearson Education, 17<sup>th</sup> edition, 2011.

**SE5013**

**MANAGING HUMAN RESOURCE**

<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

**OBJECTIVES:**

The student should be able to

- Know the importance of resources for a task.
- Compare all the resources with Human resources so the employee constraints are checked to meet the completion of the task.
- Study Training types
- Forecast Human Resource requirement.
- Know the selection procedures.

**UNIT I                    PERSPECTIVES IN HUMAN RESOURCE MANAGEMENT                    9**

Evolution of human resource management – the importance of the human factor – objectives of human resource management – role of human resource manager – human resource policies – computer applications in human resource management.

**UNIT II                    THE CONCEPT OF BEST FIT EMPLOYEE                    9**

Importance of human resource planning – forecasting human resource requirement – internal and external sources. Selection process-screening – tests - validation – interview - medical examination – recruitment introduction – importance – practices – socialization benefits.

**UNIT III                    TRAINING AND EXECUTIVE DEVELOPMENT                    9**

Types of training, methods, purpose, benefits and resistance. Executive development programmes – common practices - benefits – self-development – knowledge management.

**UNIT IV                    SUSTAINING EMPLOYEE INTEREST                    9**

Compensation plan – reward – motivation – theories of motivation – career management – development, mentor – protégé relationships.



**UNIT V PERFORMANCE EVALUATION AND CONTROL PROCESS 9**

Method of performance evaluation – feedback – industry practices. Promotion, demotion, transfer and separation – implication of job change. The control process – importance – methods – requirement of effective control systems grievances – causes – implications – redressal methods.

**TOTAL : 45 PERIODS**

**OUTCOMES:**

**At the end the student will be able to**

- Apply recruitment techniques in employee selection process.
- Apply appropriate training process
- Motivate employees by sustaining employee Interests.

**REFERENCES:**

1. BiswajeetPattanayak, "Human Resource Management", Prentice Hall of India, 2001
2. Decenzo and Robbins, "Human Resource Management", Wilsey, 6th edition, 2001
3. Dessler Human Resource Management, Pearson Education Limited, 2002
4. Human Resource Management, EugenceMckenna and Nic Beach, Pearson Education Limited, 2002
5. Ivancevich, "Human Resource Management", McGraw Hill 2002
6. Mamoria C.B. and Mamoria S, "Personnel Management", Himalaya Publishing Company, 1997
7. Mejia, Luis Gomez, David Balkin, and Robert Cardy, "Managing human resources", 7<sup>th</sup> edition, New Delhi: Prentice-Hall of India, 2012.
8. Wayne Cascio, "Managing Human Resource", McGraw Hill, 1998

<b>SE5014</b>	<b>PRINCIPLES OF SUPPLY CHAIN MANAGEMENT</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

**OBJECTIVES**

The student should be able to

- Learn about the E-business environment driven by the Automation Software in quick movement of supply of products
- Study the fundamentals of supply chain management comprising of Inventory management and warehousing etc as co parts of entire business
- Learn the cost management for the supply of products
- Improve the overall organization performance and customer satisfaction by improving product or service delivery to consumer.

**UNIT I FUNDAMENTALS OF SUPPLY CHAIN MANAGEMENT 9**

Supply chain networks, Integrated supply chain planning, Decision phases in s supply chain, process view of a supply chain, supply chain flows, Overview of supply chain models and modeling systems, Supply chain planning: Strategic, operational and tactical, Understanding supply chain through process mapping and process flow chart.

**UNIT II SCM STRATEGIES, PERFORMANCE 9**

Supply chain strategies, achieving strategic fit, value chain, Supply chain drivers and obstacles, Strategic Alliances and Outsourcing, purchasing aspects of supply chain, Supply chain performance measurement: The balanced score card approach, Performance Metrics. Planning demand and supply: Demand forecasting in supply chain, Aggregate planning in supply chain, Predictable variability.



**OBJECTIVES:****The student should be able to**

- Have an overview of the agent systems and software agents.
- Understand the basic concepts of intelligent software agents.
- Design and build a multiagent system.
- Have a basic understanding about software agent technology and to be familiar with some of the communicating languages, standardization and applications.
- Learn the use of software agents to represent and share information to coordinate activities of the agents for the purpose of group problem solving.

**UNIT I                  AGENTS – OVERVIEW    9**

Agent Definition – Agent Programming Paradigms – Agent Vs Object – Aglet – Mobile Agents – Agent Frameworks – Agent Reasoning.

**UNIT II                  JAVA AGENTS    9**

Processes – Threads – Daemons – Components – Java Beans – ActiveX – Sockets – RPCs – Distributed Computing – Aglets Programming – Jini Architecture – Actors and Agents – Typed and proactive messages.

**UNIT III                 MULTIAGENT SYSTEMS    9**

Interaction between agents – Reactive Agents – Cognitive Agents – Interaction protocols – Agent coordination – Agent negotiation – Agent Cooperation – Agent Organization – Self-Interested agents in Electronic Commerce Applications.

**UNIT IV                 INTELLIGENT SOFTWARE AGENTS    9**

Interface Agents – Agent Communication Languages – Agent Knowledge Representation – Agent Adaptability – Belief Desire Intension – Mobile Agent Applications.

**UNIT V                  AGENTS AND SECURITY    9**

Agent Security Issues – Mobile Agents Security – Protecting Agents against Malicious Hosts – Untrusted Agent – Black Box Security – Authentication for agents – Security issues for Aglets.

**TOTAL : 45 PERIODS****OUTCOMES:****At the end of this course, the students should be able to:**

- Create / develop an agent based system for a particular task.
- Design an application that uses different security issues for intelligent agents.
- Effectively apply agent-based technologies in the development and application of distributed information systems that use software agents.

**REFERENCES:**

1. Bigus&Bigus, " Constructing Intelligent agents with Java ", Wiley, 1997
2. Bradshaw, " Software Agents ", MIT Press, 2010
3. Gerhard Weiss, "Multi Agent Systems – A Modern Approach to Distributed Artificial Intelligencell", MIT Press, 2000.
4. Richard Murch, Tony Johnson, "Intelligent Software Agents", Prentice Hall, 2000
5. Russel, Norvig, "Artificial Intelligence: A Modern Approach", Second Edition, Pearson Education, 2003

SE5016

**USER INTERFACE DESIGN AND EVALUATION**

<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

**OBJECTIVES**

**The student should be able to**

- Understand how to study the tasks that the user needs to accomplish with the software system.
- Learn the constraints that affect the UI design.
- Study the importance of human- computer interaction.
- Identify the various facilities provided in WINDOWS including multimedia.

**UNIT I INTRODUCTION**

**9**

Human-Computer Interface – Characteristics Of Graphics Interface –Direct Manipulation Graphical System – Web User Interface –Popularity –Characteristic & Principles.

**UNIT II HUMAN COMPUTER INTERACTION**

**9**

User Interface Design Process – Obstacles –Usability –Human Characteristics In Design – Human Interaction Speed –Business Functions –Requirement Analysis – Direct – Indirect Methods – Basic Business Functions – Design Standards – General Design Principles – Conceptual Model Design – Conceptual Model Mock-Ups

**UNIT III WINDOWS**

**9**

Characteristics– Components– Presentation Styles– Types– Managements– Organizations– Operations– Web Systems– System Timings - Device– Based Controls Characteristics– Screen – Based Controls — Human Consideration In Screen Design – Structures Of Menu – Functions Of Menu– Contents Of Menu– Formatting – Phrasing The Menu – Selecting Menu Choice– Navigating Menus– Graphical Menus. Operate Control – Text Boxes– Selection Control– Combination Control– Custom Control– Presentation Control.

**UNIT IV MULTIMEDIA**

**9**

Text For Web Pages – Effective Feedback– Guidance & Assistance– Internationalization– Accessibility– Icons– Image– Multimedia – Coloring.

**UNIT V EVALUATION**

**9**

Conceptual Model Evaluation – Design Standards Evaluation – Detailed User Interface Design Evaluation – User centered design processes – heuristic evaluation Usability Testing – understanding users and their goals – planning for usability testing – analyzing and reporting usability test results.

**TOTAL : 45 PERIODS**

**OUTCOMES:**

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

- Design a more user friendly software.
- Utilize the existing functionalities provided and develop a better design.
- Predict the need of the end user and design the interface accordingly.

**REFERENCES:**

1. Alan Cooper, "The Essential of User Interface Design", Wiley – Dream Tech Ltd., 2002
2. Carol M. Barnum, "Usability Testing Essentials: Ready, Set...Test", Elsevier, 2010
3. Sharp, Rogers, Preece, "Interaction Design", Wiley India Edition, 2007
4. Wilbert. O. Galitz, —The Essential Guide To User Interface Design, John Wiley& Sons, 2001
5. Wilbert O. Galitz, The Essential Guide to User Interface Design – An Introduction to GUI Design Principles and techniques, 2007, wiley.

**OBJECTIVES**

- To understand the SOA architecture
- To understand the service oriented analysis and design
- To understand the development of deployment of web services
- To understand the security issues of SOA

**UNIT I SOA FUNDAMENTALS****9**

Principles of Service Orientation - Client-Server Architecture - Distributed Internet Architecture - SOA Characteristics - Anatomy of SOA - Components - Interaction - Technical and Business Benefits - Multi-channel access - Business Process Management

**UNIT II SOA AND WEB SERVICES****9**

Web Service Platform - Web Service Description - Service Contracts - Service Level Data Model - Service Discovery - Service Level Security - Service Level Interaction Patterns: SOAP basics - Messaging with SOAP - Message Exchange Patterns - Web WSDL basics, Writing a Java Web Service, writing a Java Web Service Client ,Describing Web Services: WSDL, Representing Data Types - XML Schema, Communicating Object Data, SOAP Related Technologies

**UNIT III SERVICE ORIENTED ANALYSIS AND DESIGN****9**

Design principles - Business Centric SOA - Deriving Business services - Service Modeling - Coordination - Atomic Transaction - Business activities - Web Service Orchestration Business Process Execution Language (BPEL) - Choreography - Metadata Management- Entity centric business service design - Application Service design - Task centric business service design

**UNIT IV WEB SERVICES DEVELOPMENT AND DEPLOYMENT****9**

XML and Web Services - WSDL basics - SOA support in J2EE - Java API for XML-based Web Services (JAX-WS) - Java Architecture for XML Binding (JAXB) - Java API for XML Registries (JAXR) - Web Services Interoperability Technologies - SOA support in .NET - Common Language Runtime - ASP.NET - Web forms - ASP.NET Web Services - Web Services Enhancements

**UNIT V SOA APPLICATIONS AND SECURITY****9**

Security Overview: e-commerce based security (public key cryptography) – Public key encryption – Security issues in XML document – SOAP security issue – XML Security framework: XML Digital Signature (Enveloped, enveloping and detached) – Signature validation - XML Encryption – Types – Canonicalization - XML Key management.

**TOTAL : 45 PERIODS****OUTCOMES:**

**Upon completion of this course, the student should be able to**

- Develop and deploy simple and composite web services with SOA design principles considering the security issues
- Use the standards and technologies of modern web service implementations
- Efficiently use leading development tools to create and consume web services
- Implement a service oriented application

## REFERENCES:

1. Eric Newcomer, Greg Lomow, "Understanding SOA with Web Services", Pearson Education, 2004.
2. Frank Cohen, "Fast SOA", Morgan Kaufmann, 2010.
3. Mark O' Neill, "Web Services Security", Tata McGraw-Hill Edition, 2003.
4. Sandeep Chatterjee, James Webber, "Developing Enterprise Web Services", Pearson Education, 2003.
5. Shankar Kambhampaly, "Service Oriented Architecture for Enterprise Applications", Wiley India Pvt Ltd, 2008.
6. Thomas Erl, "Service Oriented Architecture: Concepts, Technology, and Design", Pearson Education, 2006.

**MP5291**

**REAL TIME SYSTEMS**

**L T P C**  
**3 0 0 3**

### OBJECTIVES:

- To learn real time operating system concepts, the associated issues & Techniques.
- To understand design and synchronization problems in Real Time System.
- To explore the concepts of real time databases.
- To understand the evaluation techniques present in Real Time System.

### **UNIT I REAL TIME SYSTEM AND SCHEDULING 9**

Introduction– Structure of a Real Time System –Task classes – Performance Measures for Real Time Systems – Estimating Program Run Times – Issues in Real Time Computing – Task Assignment and Scheduling – Classical uniprocessor scheduling algorithms –Fault Tolerant Scheduling.

### **UNIT II SOFTWARE REQUIREMENTS ENGINEERING 9**

Requirements engineering process – types of requirements – requirements specification for real time systems – Formal methods in software specification – structured Analysis and Design – object oriented analysis and design and unified modelling language – organizing the requirements document – organizing and writing documents – requirements validation and revision.

### **UNIT III INTERTASK COMMUNICATION AND MEMORY MANAGEMENT 9**

Buffering data – Time relative Buffering- Ring Buffers – Mailboxes – Queues – Critical regions – Semaphores – other Synchronization mechanisms – deadlock – priority inversion – process stack management – run time ring buffer – maximum stack size – multiple stack arrangement – memory management in task control block - swapping – overlays – Block page management – replacement algorithms – memory locking – working sets – real time garbage collection – contiguous file systems.

### **UNIT IV REAL TIME DATABASES 9**

Real time Databases – Basic Definition, Real time Vs General Purpose Databases, Main Memory Databases, Transaction priorities, Transaction Aborts, Concurrency control issues, Disk Scheduling Algorithms, Two– phase Approach to improve Predictability – Maintaining Serialization Consistency – Databases for Hard Real Time Systems.

**UNIT V EVALUATION TECHNIQUES AND CLOCK SYNCHRONIZATION 9**

Reliability Evaluation Techniques – Obtaining parameter values, Reliability models for Hardware Redundancy–Software error models. Clock Synchronization–Clock, A Nonfault–Tolerant Synchronization Algorithm – Impact of faults – Fault Tolerant Synchronization in Hardware – Fault Tolerant Synchronization in software.

**TOTAL: 45 PERIODS**

**OUTCOMES:**

**Upon completion of this course, the students should be able to:**

- Apply principles of real time system design techniques to develop real time applications.
- Make use of database in real time applications.
- Make use of architectures and behaviour of real time operating systems.
- Apply evaluation techniques in application.

**REFERENCES:**

1. Allen Burns, Andy Wellings, “Real Time Systems and Programming Languages”, Pearson Education, 2003.
2. C.M. Krishna, Kang G. Shin, “Real-Time Systems”, McGraw-Hill International Editions, 1997
3. Philip.A.Laplante, “Real Time System Design and Analysis”, Prentice Hall of India, 3<sup>rd</sup> Edition, 2004
4. Rajib Mall, “Real-time systems: theory and practice”, Pearson Education, 2009
5. R.J.A Buhur, D.L Bailey, “An Introduction to Real-Time Systems”, Prentice Hall International, 1999
6. Stuart Bennett, “Real Time Computer Control-An Introduction”, Prentice Hall of India, 1998

**CP5291**

**SECURITY PRACTICES**

L	T	P	C
3	0	0	3

**OBJECTIVES:**

- To learn the core fundamentals of system and web security concepts
- To have through understanding in the security concepts related to networks
- To deploy the security essentials in IT Sector
- To be exposed to the concepts of Cyber Security and encryption Concepts
- To perform a detailed study of Privacy and Storage security and related Issues.

**UNIT I SYSTEM SECURITY 9**

Building a secure organization- A Cryptography primer- detecting system Intrusion- Preventing system Intrusion- Fault tolerance and Resilience in cloud computing environments- Security web applications, services and servers.

**UNIT II NETWORK SECURITY 9**

Internet Security - Botnet Problem- Intranet security- Local Area Network Security - Wireless Network Security - Wireless Sensor Network Security- Cellular Network Security- Optical Network Security- Optical wireless Security.

**UNIT III SECURITY MANEGEMENT 9**  
 Information security essentials for IT Managers- Security Management System - Policy Driven System Management- IT Security - Online Identity and User Management System - Intrusion and Detection and Prevention System.

**UNIT IV CYBER SECURITY AND CRYPTOGRAPHY 9**  
 Cyber Forensics- Cyber Forensics and Incidence Response - Security e-Discovery - Network Forensics - Data Encryption- Satellite Encryption - Password based authenticated Key establishment Protocols.

**UNIT V PRIVACY AND STORAGE SECURITY 9**  
 Privacy on the Internet - Privacy Enhancing Technologies - Personal privacy Policies - Detection of Conflicts in security policies- privacy and security in environment monitoring systems. Storage Area Network Security - Storage Area Network Security Devices - Risk management - Physical Security Essentials.

**TOTAL : 45 PERIODS**

**OUTCOMES:**

**Upon completion of this course the students should be able to**

- Understand the core fundamentals of system security
- Apply the security concepts related to networks in wired and wireless scenario
- Implement and Manage the security essentials in IT Sector
- Able to explain the concepts of Cyber Security and encryption Concepts
- Able to attain a through knowledge in the area of Privacy and Storage security and related Issues.

**REFERENCES:**

1. John R.Vacca, Computer and Information Security Handbook, Second Edition, Elsevier 2013.
2. Michael E. Whitman, Herbert J. Mattord, Principal of Information Security, Fourth Edition, Cengage Learning, 2012.
3. Richard E.Smith, Elementary Information Security, Second Edition, Jones and Bartlett Learning, 2016

<b>CP5094</b>	<b>INFORMATION RETRIEVAL TECHNIQUES</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		3	0	0	3

**OBJECTIVES:**

- To understand the basics of information retrieval with pertinence to modeling, query operations and indexing
- To get an understanding of machine learning techniques for text classification and clustering.
- To understand the various applications of information retrieval giving emphasis to multimedia IR, web search
- To understand the concepts of digital libraries



**UNIT I INTRODUCTION: MOTIVATION 9**

Basic Concepts – Practical Issues - Retrieval Process – Architecture - Boolean Retrieval –Retrieval Evaluation – Open Source IR Systems–History of Web Search – Web Characteristics–The impact of the web on IR —IR Versus Web Search–Components of a Search engine

**UNIT II MODELING 9**

Taxonomy and Characterization of IR Models – Boolean Model – Vector Model - Term Weighting – Scoring and Ranking –Language Models – Set Theoretic Models - Probabilistic Models – Algebraic Models – Structured Text Retrieval Models – Models for Browsing

**UNIT III INDEXING 9**

Static and Dynamic Inverted Indices – Index Construction and Index Compression. Searching - Sequential Searching and Pattern Matching. Query Operations -Query Languages – Query Processing - Relevance Feedback and Query Expansion - Automatic Local and Global Analysis – Measuring Effectiveness and Efficiency

**UNIT IV CLASSIFICATION AND CLUSTERING 9**

Text Classification and Naïve Bayes – Vector Space Classification – Support vector machines and Machine learning on documents. Flat Clustering – Hierarchical Clustering –Matrix decompositions and latent semantic indexing – Fusion and Meta learning

**UNIT V SEARCHING THE WEB 9**

Searching the Web –Structure of the Web –IR and web search – Static and Dynamic Ranking – Web Crawling and Indexing – Link Analysis - XML Retrieval Multimedia IR: Models and Languages – Indexing and Searching Parallel and Distributed IR – Digital Libraries

**TOTAL : 45 PERIODS**

**OUTCOMES:**

**Upon completion of this course, the student should be able to**

- Build an Information Retrieval system using the available tools
- Identify and design the various components of an Information Retrieval system
- Apply machine learning techniques to text classification and clustering which is used for efficient Information Retrieval
- Design an efficient search engine and analyze the Web content structure

**REFERENCES:**

1. Christopher D. Manning, Prabhakar Raghavan, Hinrich Schutze, “Introduction to Information Retrieval”, Cambridge University Press, First South Asian Edition, 2008.
2. Ricardo Baeza – Yates, Berthier Ribeiro – Neto, “Modern Information Retrieval: The concepts and Technology behind Search” (ACM Press Books), Second Edition, 2011.
3. Stefan Buttcher, Charles L. A. Clarke, Gordon V. Cormack, “Information Retrieval Implementing and Evaluating Search Engines”, The MIT Press, Cambridge, Massachusetts London, England, 2010.

**OBJECTIVES :****The students should be made to:**

- Learn design challenges and design methodologies
- Study general and single purpose processor
- Understand bus structures

**UNIT I EMBEDDED SYSTEM OVERVIEW 9**

Embedded System Overview, Design Challenges – Optimizing Design Metrics, Design Methodology, RT-Level Combinational and Sequential Components, Optimizing Custom Single-Purpose Processors.

**UNIT II GENERAL AND SINGLE PURPOSE PROCESSOR 9**

Basic Architecture, Pipelining, Superscalar and VLIW architectures, Programmer's view, Development Environment, Application-Specific Instruction-Set Processors (ASIPs) Microcontrollers, Timers, Counters and watchdog Timer, UART, LCD Controllers and Analog-to-Digital Converters, Memory Concepts.

**UNIT III BUS STRUCTURES 9**

Basic Protocol Concepts, Microprocessor Interfacing – I/O Addressing, Port and Bus-Based I/O, Arbitration, Serial Protocols, I<sup>2</sup>C, CAN and USB, Parallel Protocols – PCI and ARM Bus, Wireless Protocols – IrDA, Bluetooth, IEEE 802.11.

**UNIT IV STATE MACHINE AND CONCURRENT PROCESS MODELS 9**

Basic State Machine Model, Finite-State Machine with Datapath Model, Capturing State Machine in Sequential Programming Language, Program-State Machine Model, Concurrent Process Model, Communication among Processes, Synchronization among processes, Dataflow Model, Real-time Systems, Automation: Synthesis, Verification : Hardware/Software Co-Simulation, Reuse: Intellectual Property Cores, Design Process Models.

**UNIT V EMBEDDED SOFTWARE DEVELOPMENT TOOLS AND RTOS 9**

Compilation Process – Libraries – Porting kernels – C extensions for embedded systems – emulation and debugging techniques – RTOS – System design using RTOS.

**TOTAL: 45 PERIODS****OUTCOMES:****At the end of this course, the students should be able to:**

- Explain different protocols
- Discuss state machine and design process models
- Outline embedded software development tools and RTOS

**REFERENCES:**

1. Bruce Powel Douglas, "Real time UML, second edition: Developing efficient objects for embedded systems", 3rd Edition 1999, Pearson Education.
2. Daniel W. Lewis, "Fundamentals of embedded software where C and assembly meet", Pearson Education, 2002.
3. Frank Vahid and Tony Gwargie, "Embedded System Design", John Wiley & sons, 2002.
4. Steve Heath, "Embedded System Design", Elsevier, Second Edition, 2004.