

ANNA UNIVERSITY, CHENNAI
NON - AUTONOMOUS COLLEGES AFFILIATED ANNA UNIVERSITY
M.E. SOFTWARE ENGINEERING
REGULATIONS – 2021
CHOICE BASED CREDIT SYSTEM

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs):

- I. Use all the phases of the Software Development Life Cycle to build robust software systems
- II. Analyze and evaluate problems critically using the theoretical and technical knowledge to develop sustainable solutions and systems
- III. Identify the requirements and implement reliable, innovative and appropriate software solutions for the industrial need
- IV. Enhance skills through lifelong learning as software professionals to progress in managerial and leadership roles.
- V. Work efficiently in multidisciplinary teams with effective communication and follow ethical principles.

PROGRAMME OUTCOMES

1. An ability to independently carry out research/investigation and development work to solve practical problems
 2. An ability to write and present a substantial technical report/document
- Students should be able to demonstrate a degree of mastery over the area as per the specialization of the program. The mastery should be at a level higher than the requirements in the appropriate bachelor program
- 3.
 4. Collect requirements from the stakeholders and design software engineering applications with deep understanding of best software principles and practices.
 5. Apply software testing techniques to produce error free and reliable software and ensure quality.
 6. Manage software project with state of the art approaches to ensure balance in all project areas like time, cost, quality, risk and human resource.
-

ANNA UNIVERSITY, CHENNAI
NON - AUTONOMOUS COLLEGES AFFILIATED ANNA UNIVERSITY
M.E. SOFTWARE ENGINEERING
REGULATIONS – 2021
CHOICE BASED CREDIT SYSTEM
I TO IV SEMESTERS CURRICULA AND 1st SEMESTER SYLLABI
SEMESTER I

S. NO.	COURSE CODE	COURSE TITLE	CATE-GORY	PERIODS PER WEEK			TOTAL CONTACT PERIODS	CREDITS
				L	T	P		
THEORY								
1.	MA4157	Mathematical Modeling and Simulation	FC	4	0	0	4	4
2.	RM4151	Research Methodology and IPR	RMC	2	0	0	2	2
3.	CP4151	Advanced Data Structures and Algorithms	PCC	3	0	0	3	3
4.	CP4152	Database Practices	PCC	3	0	2	5	4
5.	SE4151	Advanced Software Engineering	PCC	3	0	0	3	3
6.	SE4101	Software Architecture	PCC	3	0	0	3	3
7.		Audit Course – I*	AC	2	0	0	2	0
PRACTICALS								
8.	CP4161	Advanced Data Structures and Algorithms Laboratory	PCC	0	0	4	4	2
9.	SE4111	Advanced Software Tools Laboratory	PCC	0	0	4	4	2
TOTAL				20	0	10	30	23

*Audit course is optional

SEMESTER II

S. NO.	COURSE CODE	COURSE TITLE	CATE-GORY	PERIODS PER WEEK			TOTAL CONTACT PERIODS	CREDITS
				L	T	P		
THEORY								
1.	SE4201	Software Requirements Engineering	PCC	3	0	0	3	3
2.	SE4202	Software System Design	PCC	3	0	0	3	3
3.	SE4203	Software Testing	PCC	3	0	0	3	3
4.	SE4204	Integrated Software Project Management	PCC	3	0	0	3	3
5.		Professional Elective I	PEC	3	0	0	3	3
6.		Professional Elective II	PEC	3	0	0	3	3
7.		Audit Course – II*	AC	2	0	0	2	0
PRACTICALS								
8.	SE4211	Term Paper Writing and seminar	EEC	0	0	2	2	1
9.	SE4212	Software Development Laboratory	PCC	0	0	4	4	2
TOTAL				20	0	6	26	21

*Audit course is optional

SEMESTER III

S. NO.	COURSE CODE	COURSE TITLE	CATE-GORY	PERIODS PER WEEK			TOTAL CONTACT PERIODS	CREDITS
				L	T	P		
THEORY								
1.	SE4301	Software Reliability and Quality	PCC	3	0	0	3	3
2.		Professional Elective III	PEC	3	0	0	3	3
3.		Professional Elective IV	PEC	3	0	2	5	4
4.		Open Elective	OEC	3	0	0	3	3
PRACTICALS								
5.	SE4311	Project Work I	EEC	0	0	12	12	6
TOTAL				12	0	14	26	19

SEMESTER IV

S. NO.	COURSE CODE	COURSE TITLE	CATE-GORY	PERIODS PER WEEK			TOTAL CONTACT PERIODS	CREDITS
				L	T	P		
PRACTICALS								
6.	SE4411	Project Work II	EEC	0	0	24	24	12
TOTAL				0	0	24	24	12

TOTAL NO. OF CREDITS: 75

**PROFESSIONAL ELECTIVES
SEMESTER II, ELECTIVE I**

S. NO.	COURSE CODE	COURSE TITLE	CATE-GORY	PERIODS PER WEEK			TOTAL CONTACT PERIODS	CREDITS
				L	T	P		
1.	SE4071	Agile Methodologies	PEC	3	0	0	3	3
2.	IF4095	Social Network Analysis	PEC	3	0	0	3	3
3.	BC4152	Cyber Forensics and Investigation	PEC	3	0	0	3	3
4.	MP4251	Cloud Computing Technologies	PEC	3	0	0	3	3

SEMESTER II, ELECTIVE II

S. NO.	COURSE CODE	COURSE TITLE	CATE-GORY	PERIODS PER WEEK			TOTAL CONTACT PERIODS	CREDITS
				L	T	P		
1.	SE4072	Image Processing	PEC	3	0	0	3	3
2.	CP4093	Information Retrieval Techniques	PEC	3	0	0	3	3
3.	MP4091	Cognitive Computing	PEC	3	0	0	3	3
4.	IF4094	Pattern Recognition	PEC	3	0	0	3	3
5.	BD4251	Big Data Mining and Analytics	PEC	3	0	0	3	3
6.	BD4151	Foundations of Data Science	PEC	3	0	0	3	3

SEMESTER III, ELECTIVE III

S. NO.	COURSE CODE	COURSE TITLE	CATEGORY	PERIODS PER WEEK			TOTAL CONTACT PERIODS	CREDITS
				L	T	P		
1.	IF4072	Design Thinking	PEC	3	0	0	3	3
2.	IF4093	GPU Computing	PEC	3	0	0	3	3
3.	MP4094	Web Services and API Design	PEC	3	0	0	3	3

SEMESTER III, ELECTIVE IV

S. NO.	COURSE CODE	COURSE TITLE	CATEGORY	PERIODS PER WEEK			TOTAL CONTACT PERIODS	CREDITS
				L	T	P		
1.	IF4073	Devops and Microservices	PEC	3	0	2	5	4
2.	IF4071	Deep Learning	PEC	3	0	2	5	4
3.	CP4072	Blockchain Technologies	PEC	3	0	2	5	4
4.	IF4291	Full Stack Web Application Development	PEC	3	0	2	5	4
5.	SE4073	Embedded Software Development	PEC	3	0	2	5	4
6.	IF4074	Distributed Application Development	PEC	3	0	2	5	4
7.	CP4291	Internet of Things	PEC	3	0	2	5	4
8.	MU4291	Mixed Reality	PEC	3	0	2	5	4

AUDIT COURSES (AC)

Registration for any of these courses is optional to students

SL. NO	COURSE CODE	COURSE TITLE	PERIODS PER WEEK			CREDITS
			L	T	P	
1.	AX4091	English for Research Paper Writing	2	0	0	0
2.	AX4092	Disaster Management	2	0	0	0
3.	AX4093	Constitution of India	2	0	0	0
4.	AX4094	நற்றமிழ் இலக்கியம்	2	0	0	0

FOUNDATION COURSES (FC)

S. NO	COURSE CODE	COURSE TITLE	PERIODS PER WEEK			CREDITS	SEMESTER
			Lecture	Tutorial	Practical		
1.	MA4157	Mathematical Modeling and Simulation	4	0	0	4	I

PROFESSIONAL CORE COURSES (PCC)

S. NO	COURSE CODE	COURSE TITLE	PERIODS PER WEEK			CREDITS	SEMESTER
			Lecture	Tutorial	Practical		
1.	CP4151	Advanced Data Structures and Algorithms	3	0	0	3	I
2.	CP4152	Database Practices	3	0	2	4	I
3.	SE4151	Advanced Software	3	0	0	3	I
4.	SE4101	Software Architecture	3	0	0	3	I
5.	CP4161	Advanced Data Structures and Algorithms Laboratory	0	0	4	2	I
6.	SE4111	Advanced Software Tools Laboratory	0	0	4	2	I
7.	SE4201	Software Requirements Engineering	3	0	0	3	II
8.	SE4202	Software System Design	3	0	0	3	II
9.	SE4203	Software Testing	3	0	0	3	II
10.	SE4204	Integrated Software Project Management	3	0	0	3	II
11.	SE4212	Software Development Laboratory	0	0	4	2	II
12.	SE4301	Software Reliability and Quality	3	0	0	3	III

RESEARCH METHODOLOGY AND IPR COURSES (RMC)

S. NO	COURSE CODE	COURSE TITLE	PERIODS PER WEEK			CREDITS	SEMESTER
			Lecture	Tutorial	Practical		
1.	RM4151	Research Methodology and IPR	2	0	0	2	1

EMPLOYABILITY ENHANCEMENT COURSES (EEC)

S. NO	COURSE CODE	COURSE TITLE	PERIODS PER WEEK			CREDITS	SEMESTER
			Lecture	Tutorial	Practical		
1.	SE4211	Term Paper Writing and seminar	0	0	2	1	II
2.	SE 4311	Project Work I	0	0	12	6	III
3.	SE 4411	Project Work II	0	0	24	12	IV

SUMMARY

Sl. No.	NAME OF THE PROGRAMME: M.E. SOFTWARE ENGINEERING					
	SUBJECT AREA	CREDITS PER SEMESTER				CREDITS TOTAL
		I	II	III	IV	
1.	FC	04	00	00	00	04
2.	PCC	17	14	03	00	34
3.	PEC	00	06	07	00	13
4.	RMC	02	00	00	00	02
5.	OEC	00	00	03	00	03
6.	EEC	00	01	06	12	19
7.	Non Credit/Audit Course	✓	✓	00	00	
8.	TOTAL CREDIT	23	21	19	12	75



COURSE OBJECTIVES:

This course will help the students to

- acquire the knowledge of solving system of linear equations using an appropriate numerical methods.
- approximate the functions using polynomial interpolation numerical differentiation and integration using interpolating polynomials.
- acquire the knowledge of numerical solution of ordinary differential equation by single and multi step methods.
- obtain the solution of boundary value problems in partial differential equations using finite differences.
- study simulation and Monte-Carlo methods and their applications.

UNIT I MATRICES AND LINEAR SYSTEMS OF EQUATIONS 12

Solution of Linear Systems : Cramer's Rule - Gaussian elimination and Gauss Jordan methods - Cholesky decomposition method – Gauss Seidel iteration method - Eigenvalue problems : Power method with deflation for both symmetric and non symmetric matrices and Jacobi method for symmetric matrices.

UNIT II INTERPOLATION, DIFFERENTIATION AND INTEGRATION 12

Lagrange's interpolation - Newton's divided differences - Hermite's interpolation – Newton's forward and backward differences – Numerical differentiation – Numerical integration : Trapezoidal and Simpson's $\frac{1}{3}$ rules - Gaussian quadrature : 2 and 3 point rules.

UNIT III DIFFERENTIAL EQUATIONS 12

Initial value problems for first and second order ODEs : Single step methods - Taylor's series method – Euler's and modified Euler's methods - Runge - Kutta method of fourth order - Multi step methods : Milne's and Adam Bashforth methods - Boundary value problems : Finite difference approximations to derivatives - Finite difference method of solving second order ODEs .

UNIT IV PARTIAL DIFFERENTIAL EQUATIONS 12

Classification of second order PDE's - Finite difference approximations to partial derivatives - Elliptic equations : Solution of Laplace and Poisson equations - One dimensional parabolic equation - Bender Schmidt method - Hyperbolic equation : One dimensional wave equation.

UNIT V SIMULATION AND MONTE CARLO METHODS 12

Random numbers : Random number algorithms and generators – Estimation of areas and volumes by Monte Carlo techniques - Numerical integration - Computing volumes – Simulation : Loaded Die Problem - Birthday problem - Buffon's needle problem - Two dice problem and Neutron shielding problem.

TOTAL: 60 PERIODS

COURSE OUTCOMES :

At the end of the course, students will be able to

- solve an algebraic or transcendental equation and linear system of equations using an appropriate numerical method.
- approximation of functions using polynomial interpolation, numerical differentiation and integration using interpolating polynomials.
- numerical solution of differential equations by single and multistep methods.

- solution of boundary value problems and initial boundary value problems in partial differential equations using finite differences.
- simulation and Monte-Carlo methods and their applications.

REFERENCES :

1. Burden, R.L. and Faires, J.D. "Numerical Analysis", 9th Edition, Cengage Learning, Delhi, 2016.
2. Cheney, W and Kincaid D., "Numerical Mathematics and Computing", 7th Edition, Cengage Learning, Delhi, 2014.
3. Jain, M.K., Iyengar, S.R.K. and Jain R.K. "Numerical Methods for Scientific and Engineering Computation", 6th Edition, New Age International Pvt. Ltd., Delhi, 2014.
4. Landau, D.P. and Binder, K., "A Guide to Monte - Carlo Simulations in Statistical Physics", 3rd Edition, Cambridge University Press, Cambridge, 2009.
5. Maki, D P and Thompson, M., "Mathematical Modelling with Computer Simulation", Cengage Learning, Delhi, 2011.
6. Sastry, S.S., "Introductory Methods of Numerical Analysis", 5th Edition, PHI Learning Pvt. Ltd., Delhi, 2012.
7. Taha, H.A. "Operations Research", 10th Edition, Pearson Education India, Delhi, 2018.

RM4151	RESEARCH METHODOLOGY AND IPR	L T P C
		2 0 0 2
UNIT I	RESEARCH DESIGN	6
Overview of research process and design, Use of Secondary and exploratory data to answer the research question, Qualitative research, Observation studies, Experiments and Surveys.		
UNIT II	DATA COLLECTION AND SOURCES	6
Measurements, Measurement Scales, Questionnaires and Instruments, Sampling and methods. Data - Preparing, Exploring, examining and displaying.		
UNIT III	DATA ANALYSIS AND REPORTING	6
Overview of Multivariate analysis, Hypotheses testing and Measures of Association. Presenting Insights and findings using written reports and oral presentation.		
UNIT IV	INTELLECTUAL PROPERTY RIGHTS	6
Intellectual Property – The concept of IPR, Evolution and development of concept of IPR, IPR development process, Trade secrets, utility Models, IPR & Bio diversity, Role of WIPO and WTO in IPR establishments, Right of Property, Common rules of IPR practices, Types and Features of IPR Agreement, Trademark, Functions of UNESCO in IPR maintenance.		
UNIT V	PATENTS	6
Patents – objectives and benefits of patent, Concept, features of patent, Inventive step, Specification, Types of patent application, process E-filing, Examination of patent, Grant of patent, Revocation, Equitable Assignments, Licences, Licensing of related patents, patent agents, Registration of patent agents.		

TOTAL: 30 PERIODS

REFERENCES:

1. Cooper Donald R, Schindler Pamela S and Sharma JK, "Business Research Methods", Tata McGraw Hill Education, 11e (2012).
2. Catherine J. Holland, "Intellectual property: Patents, Trademarks, Copyrights, Trade Secrets", Entrepreneur Press, 2007.
3. David Hunt, Long Nguyen, Matthew Rodgers, "Patent searching: tools & techniques", Wiley, 2007.
4. The Institute of Company Secretaries of India, Statutory body under an Act of parliament, "Professional Programme Intellectual Property Rights, Law and practice", September 2013.

CP4151

ADVANCED DATA STRUCTURES AND ALGORITHMS

L T P C

3 0 0 3

COURSE 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 & COMPLEXITY ANALYSIS**

9

Algorithms – Algorithms as a Technology -Time and Space complexity of algorithms- Asymptotic analysis-Average and worst-case analysis-Asymptotic notation-Importance of efficient algorithms- Program performance measurement - Recurrences: The Substitution Method – The Recursion-Tree Method- Data structures and algorithms.

UNIT II **HIERARCHICAL DATA STRUCTURES**

9

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- Heap – Heap Implementation – Disjoint Sets - Fibonacci Heaps: structure – Mergeable-heap operations- Decreasing a key and deleting a node-Bounding the maximum degree.

UNIT III **GRAPHS**

9

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; Dynamic Programming - All-Pairs Shortest Paths: Shortest Paths and Matrix Multiplication – The Floyd-Warshall Algorithm

UNIT IV **ALGORITHM DESIGN TECHNIQUES**

9

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

UNIT V NP COMPLETE AND NP HARD

9

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

TOTAL: 45 PERIODS

SUGGESTED ACTIVITIES:

1. Write an algorithm for Towers of Hanoi problem using recursion and analyze the complexity (No of disc-4)
2. Write any one real time application of hierarchical data structure
3. Write a program to implement Make_Set, Find_Set and Union functions for Disjoint Set Data Structure for a given undirected graph $G(V,E)$ using the linked list representation with simple implementation of Union operation
4. Find the minimum cost to reach last cell of the matrix from its first cell
5. Discuss about any NP completeness problem

COURSE OUTCOMES:

CO1: Design data structures and algorithms to solve computing problems.

CO2: Choose and implement efficient data structures and apply them to solve problems.

CO3: Design algorithms using graph structure and various string-matching algorithms to solve real-life problems.

CO4: Design one's own algorithm for an unknown problem.

CO5: Apply suitable design strategy for problem solving.

REFERENCES

1. S.Sridhar," Design and Analysis of Algorithms", Oxford University Press, 1st Edition, 2014.
2. Adam Drozdex, "Data Structures and algorithms in C++", Cengage Learning, 4th Edition, 2013.
3. T.H. Cormen, C.E.Leiserson, R.L. Rivest and C.Stein, "Introduction to Algorithms", Prentice Hall of India, 3rd Edition, 2012.
4. Mark Allen Weiss, "Data Structures and Algorithms in C++", Pearson Education, 3rd Edition, 2009.
5. E. Horowitz, S. Sahni and S. Rajasekaran, "Fundamentals of Computer Algorithms", University Press, 2nd Edition, 2008.
6. Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, "Data Structures and Algorithms", Pearson Education, Reprint 2006.

CP4152

DATABASE PRACTICES

L T P C

3 0 2 4

COURSE OBJECTIVES:

- Describe the fundamental elements of relational database management systems
- Explain the basic concepts of relational data model, entity-relationship model, relational database design, relational algebra and SQL.
- Understand query processing in a distributed database system
- Understand the basics of XML and create well-formed and valid XML documents.
- Distinguish the different types of NoSQL databases
- To understand the different models involved in database security and their applications in real time world to protect the database and information associated with them.

UNIT I RELATIONAL DATA MODEL

15

Entity Relationship Model – Relational Data Model – Mapping Entity Relationship Model to Relational Model – Relational Algebra – Structured Query Language – Database Normalization.

Suggested Activities:

Data Definition Language

- Create, Alter and Drop
- Enforce Primary Key, Foreign Key, Check, Unique and Not Null Constraints
- Creating Views

Data Manipulation Language

- Insert, Delete, Update
- Cartesian Product, Equi Join, Left Outer Join, Right Outer Join and Full Outer Join
- Aggregate Functions
- Set Operations
- Nested Queries

Transaction Control Language

- Commit, Rollback and Save Points

UNIT II DISTRIBUTED DATABASES, ACTIVE DATABASES AND OPEN DATABASE CONNECTIVITY

15

Distributed Database Architecture – Distributed Data Storage – Distributed Transactions – Distributed Query Processing – Distributed Transaction Management – Event Condition Action Model – Design and Implementation Issues for Active Databases – Open Database Connectivity.

Suggested Activities:

- Distributed Database Design and Implementation
- Row Level and Statement Level Triggers
- Accessing a Relational Database using PHP, Python and R

UNIT III XML DATABASES

15

Structured, Semi structured, and Unstructured Data – XML Hierarchical Data Model – XML Documents – Document Type Definition – XML Schema – XML Documents and Databases – XML Querying – XPath – XQuery

Suggested Activities:

- Creating XML Documents, Document Type Definition and XML Schema
- Using a Relational Database to store the XML documents as text
- Using a Relational Database to store the XML documents as data elements
- Creating or publishing customized XML documents from pre-existing relational databases
- Extracting XML Documents from Relational Databases
- XML Querying

UNIT IV NOSQL DATABASES AND BIG DATA STORAGE SYSTEMS

15

NoSQL – Categories of NoSQL Systems – CAP Theorem – Document-Based NoSQL Systems and MongoDB – MongoDB Data Model – MongoDB Distributed Systems Characteristics – NoSQL Key-Value Stores – DynamoDB Overview – Voldemort Key-Value Distributed Data Store – Wide Column NoSQL Systems – Hbase Data Model – Hbase Crud Operations – Hbase Storage and

Distributed System Concepts – NoSQL Graph Databases and Neo4j – Cypher Query Language of Neo4j – Big Data – MapReduce – Hadoop – YARN.

Suggested Activities:

- Creating Databases using MongoDB, DynamoDB, Voldemort Key-Value Distributed Data Store Hbase and Neo4j.
- Writing simple queries to access databases created using MongoDB, DynamoDB, Voldemort Key-Value Distributed Data Store Hbase and Neo4j.

UNIT V DATABASE SECURITY

15

Database Security Issues – Discretionary Access Control Based on Granting and Revoking Privileges – Mandatory Access Control and Role-Based Access Control for Multilevel Security – SQL Injection – Statistical Database Security – Flow Control – Encryption and Public Key Infrastructures – Preserving Data Privacy – Challenges to Maintaining Database Security – Database Survivability – Oracle Label-Based Security.

Suggested Activities:

Implementing Access Control in Relational Databases

TOTAL: 75 PERIODS

COURSE OUTCOMES:

At the end of the course, the students will be able to

- Convert the ER-model to relational tables, populate relational database and formulate SQL queries on data.
- Understand and write well-formed XML documents
- Be able to apply methods and techniques for distributed query processing.
- Design and Implement secure database systems.
- Use the data control, definition, and manipulation languages of the NoSQL databases

REFERENCES:

1. R. Elmasri, S.B. Navathe, "Fundamentals of Database Systems", Seventh Edition, Pearson Education 2016.
2. Henry F Korth, Abraham Silberschatz, S. Sudharshan, "Database System Concepts", Seventh Edition, McGraw Hill, 2019.
3. C.J.Date, A.Kannan, S.Swamynathan, "An Introduction to Database Systems, Eighth Edition, Pearson Education, 2006
4. Raghu Ramakrishnan , Johannes Gehrke "Database Management Systems", Fourth Edition, McGraw Hill Education, 2015.
5. Harrison, Guy, "Next Generation Databases, NoSQL and Big Data" , First Edition, Apress publishers, 2015
6. Thomas Cannolly and Carolyn Begg, "Database Systems, A Practical Approach to Design, Implementation and Management", Sixth Edition, Pearson Education, 2015

COURSE OBJECTIVES:

- To understand the rationale for software development process models
- To understand why the architectural design of software is important;
- To understand the five important dimensions of dependability, namely, availability, reliability, safety, security, and resilience.
- To understand the basic notions of a web service, web service standards, and service-oriented architecture;
- To understand the different stages of testing from testing during development of a software system

UNIT I SOFTWARE PROCESS & MODELING 9

Prescriptive Process Models – Agility and Process – Scrum – XP – Kanban – DevOps – Prototype Construction – Prototype Evaluation – Prototype Evolution – Modeling – Principles – Requirements Engineering – Scenario-based Modeling – Class-based Modeling – Functional Modeling – Behavioral Modeling.

UNIT II SOFTWARE DESIGN 9

Design Concepts – Design Model – Software Architecture – Architectural Styles – Architectural Design – Component-Level Design – User Experience Design – Design for Mobility – Pattern-Based Design.

UNIT III SYSTEM DEPENDABILITY AND SECURITY 9

Dependable Systems – Dependability Properties – Sociotechnical Systems – Redundancy and Diversity – Dependable Processes – Formal Methods and Dependability – Reliability Engineering – Availability and Reliability – Reliability Requirements – Fault-tolerant Architectures – Programming for Reliability – Reliability Measurement – Safety Engineering – Safety-critical Systems – Safety Requirements – Safety Engineering Processes – Safety Cases – Security Engineering – Security and Dependability – Safety and Organizations – Security Requirements – Secure System Design – Security Testing and Assurance – Resilience Engineering – Cybersecurity – Sociotechnical Resilience – Resilient Systems Design.

UNIT IV SERVICE-ORIENTED SOFTWARE ENGINEERING, SYSTEMS ENGINEERING AND REAL-TIME SOFTWARE ENGINEERING 9

Service-oriented Architecture – RESTful Services – Service Engineering – Service Composition – Systems Engineering – Sociotechnical Systems – Conceptual Design – System Procurement – System Development – System Operation and Evolution – Real-time Software Engineering – Embedded System Design – Architectural Patterns for Real-time Software – Timing Analysis – Real-time Operating Systems.

UNIT V SOFTWARE TESTING AND SOFTWARE CONFIGURATION MANAGEMENT 9

Software Testing Strategy – Unit Testing – Integration Testing – Validation Testing – System Testing – Debugging – White-Box Testing – Basis Path Testing – Control Structure Testing – Black-Box Testing – Software Configuration Management (SCM) – SCM Repository – SCM Process – Configuration Management for Web and Mobile Apps.

SUGGESTED ACTIVITIES

1. Comparatively analyzing different Agile methodologies.

2. Describing the scenarios where 'Scrum' and 'Kanban' are used.
3. Mapping the data flow into suitable software architecture.
4. Developing behavioural representations for a class or component.
5. Implementing simple applications as RESTful service.

TOTAL: 45 PERIODS

COURSE OUTCOMES:

The Students will be able to

CO1:Identify appropriate process models based on the Project requirements

CO2:Understand the importance of having a good Software Architecture.

CO3:Understand the five important dimensions of dependability, namely, availability, reliability, safety, security, and resilience.

CO4:Understand the basic notions of a web service, web service standards, and service-oriented architecture;

CO5:Be familiar with various levels of Software testing

REFERENCES:

1. Software Engineering: A Practitioner's Approach, 9th Edition. Roger Pressman and Bruce Maxim, McGraw-Hill 2019.
2. Software Engineering, 10th Edition, Ian Somerville, Pearson Education Asia 2016.
3. Software Architecture In Practice, 3rd Edition, Len Bass, Paul Clements and Rick Kazman, Pearson India 2018
4. An integrated approach to Software Engineering, 3rd Edition, Pankaj Jalote, Narosa Publishing House, 2018
5. Fundamentals of Software Engineering, 5th Edition, Rajib Mall, PHI Learning Private Ltd, 2018

SE4101

SOFTWARE ARCHITECTURE

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

COURSE OBJECTIVES

- Understand the fundamentals of software architecture.
- Study the various software modeling techniques.
- Understand software implementation and deployment
- Learn the architecture of different applications.
- Relate software architecture and software quality.

UNIT I BASIC CONCEPTS, DESIGNING ARCHITECTURES

9

Understanding Software Architecture-Concepts of Software Architecture, Models, Processes, Stakeholders, the Design Process, Architectural Conception, Refined Experience in Action: Styles and Architectural Patterns, Architectural Conception in Absence of Experience

Suggested Activities

1. Identifying the pitfalls that are likely to occur for software architecture teams
2. Discussing about the role of Software architects as Cost estimators

UNIT II CONNECTORS, MODELING

9

Connectors in Action: A Motivating Example, Connector Foundations, Connector Roles, Connector Types and Their Variation Dimensions, Example Connectors, Modeling Concepts, Ambiguity,

Accuracy, and Precision, Complex Modeling: Mixed Content and Multiple Views, Evaluating Modeling Techniques, Specific Modeling Techniques

1. Identifying a few commercial products which are infrastructure components that provide asynchronous messaging service.
2. Finding out the different possibilities of splitting the system into a number of computationally independent execution structures

UNIT III ANALYSIS, IMPLEMENTATION AND DEPLOYMENT 9

Concepts, Existing Frameworks, Software Architecture and Deployment, Software Architecture and Mobility, Pipes and Filters, Event- based, Implicit Invocation, Layered systems, Repositories Interpreters, Process control

Suggested Activities

1. Identifying the type of a given architectural pattern.
2. Representing software using pipe-filter architecture.

UNIT IV APPLIED ARCHITECTURES AND STYLES 9

Distributed and Networked Architectures, Architectures for Network-Based Applications, Decentralized Architectures, Service-Oriented Architectures and Web Services, Efficiency, Complexity, Scalability and Heterogeneity, Adaptability, Dependability.

Suggested Activities

1. Identifying functional aspects of a service oriented architecture.
2. Discussing the pros and cons of implementing a middle ware to deal with architectural issues.

UNIT V IMPLEMENTATION 9

Understanding quality attributes- Availability- Deployability- Working with Other Quality Attributes- Virtualization- the Cloud and Distributed Computing- Architecturally Significant Requirements- Designing an Architecture

Suggested Activity

1. Identifying the cost of modifications in projects that measure deployment separately.
2. "Using the cloud assumes your application is service oriented." Find some examples that would support that statement and, if it is not universally true, find some that would falsify it.

TOTAL :45 PERIODS

Suggested Activity:

Students may be given problem domain that they may be encouraged to come out with multiple solution domains by applying some pattern. The best solution would be selected and presented.

COURSE OUTCOMES:

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

- Develop Software applications starting from software architecture and design.
- Apply different types of systems analysis techniques and software design strategies.
- Learn to implement and deploy software applications.
- Evaluate and implement different types of design patterns based on the requirement and functionality

- Evaluate the quality attributes for software architecture

REFERENCES:

1. Richard N. Taylor, Nenad Medvidovic, Eric Dashofy, Software Architecture: Foundations, Theory, and Practice, 2009.
2. Steven John Metsker, "Design Pattern Jav3a Workbook", Addison Wesley Workbook", 2002
3. Len Bass, Paul Clements, Rick Kazman: Software Architecture in Practice, Pearson, 4th Edition, 2021.
4. M. Shaw and D Garlan : Software Architecture Perspectives on an Emerging Discipline, Prentice- Hall. Digitized version 2007
5. Mark Richards, Neal Ford, " Fundamentals of Software Architecture, An Engineering Approach", O'Reilly Media 2020

CP4161

ADVANCED DATA STRUCTURES AND ALGORITHMS LABORATORY

L T P C
0 0 4 2

COURSE 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
- To understand the problems such as matrix chain multiplication, activity selection and Huffman coding
- To understand the necessary mathematical abstraction to solve problems.

LIST OF EXPERIMENTS:

- 1: Implementation of recursive function for tree traversal and Fibonacci
- 2: Implementation of iteration function for tree traversal and Fibonacci
- 3: Implementation of Merge Sort and Quick Sort
- 4: Implementation of a Binary Search Tree
- 5: Red-Black Tree Implementation
- 6: Heap Implementation
- 7: Fibonacci Heap Implementation
- 8: Graph Traversals
- 9: Spanning Tree Implementation
- 10: Shortest Path Algorithms (Dijkstra's algorithm, Bellman Ford Algorithm)
- 11: Implementation of Matrix Chain Multiplication
- 12: Activity Selection and Huffman Coding Implementation

HARDWARE/SOFTWARE REQUIREMENTS

- 1: 64-bit Open source Linux or its derivative
- 2: Open Source C++ Programming tool like G++/GCC

COURSE OUTCOMES:

- CO1:** Design and implement basic and advanced data structures extensively
- CO2:** Design algorithms using graph structures
- CO3:** Design and develop efficient algorithms with minimum complexity using design

techniques

CO4: Develop programs using various algorithms.

CO5: Choose appropriate data structures and algorithms, understand the ADT/libraries, and use it to design algorithms for a specific problem.

TOTAL: 60 PERIODS

REFERENCES

1. Lipschutz Seymour, "Data Structures Schaum's Outlines Series", Tata McGraw Hill, 3rd Edition, 2014.
2. Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, "Data Structures and Algorithms", Pearson Education, Reprint 2006.
3. <http://www.coursera.org/specializations/data-structures-algorithms>
4. http://www.tutorialspoint.com/data_structures_algorithms
5. <http://www.geeksforgeeks.org/data-structures/>

SE4111

ADVANCED SOFTWARE TOOLS LABORATORY

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

COURSE OBJECTIVES

- To understand the software development process, methodologies and work flow
- To be familiar with all the UML notations and understand how it supports the entire software development process
- To understand how to map a design to code and code to a good design.
- To apply Black box and White box strategies to design test cases.
- To be familiar with the modern Computer aided Software Engineering tools

Identify ambiguities, inconsistencies and incompleteness from a requirements specification and state functional and non-functional requirement

Identify different actors and use cases from a given problem statement and draw use case diagram to associate use cases with different types of relationship

Draw a class diagram after identifying classes and association among them

Graphically represent various UML diagrams, and associations among them and identify the logical sequence of activities undergoing in a system, and represent them pictorially

Able to use modern engineering tools for specification, design, implementation and testing

1. Prepare Software Requirements Specification Document.
2. Map the Software Requirements Specification Document to Design Document.
3. Model Entity Relationship Diagram and Map the Entity Relationship Diagram to Relations.
4. Model Data Flow Diagram and Map the Data Flow Diagram to Modular Design.

Using an UML Tool Perform the Following:

1. Model a Class Diagram and Map the Class Diagram to Code.
2. Model Use Case Diagrams and Sequence Diagrams.
3. Model a State Transition Diagram.
4. Model an Activity diagram
5. Model a Collaboration Diagram
6. Model a Component diagram
7. Model a Deployment Diagram
8. Generating Code from UML Models (Forward Engineering)
9. Generating UML Models from Code (Reverse Engineering)

10. Version Control configuration and use.
11. Designing Test Suites.
12. Estimation of Test Coverage Metrics and Structural Complexity
13. Unit testing using JUnit.
14. Web Application testing using Selenium
15. Test management using any open source tool
16. Mini project strictly following a Software Development Life Cycle

TOTAL: 60 PERIODS

COURSE OUTCOMES

Upon completion of course, students will be able to

- Identify ambiguities, inconsistencies and incompleteness from a requirements specification and state functional and non-functional requirement
- Identify different actors and use cases from a given problem statement and draw use case diagram to associate use cases with different types of relationship
- Graphically represent various UML diagrams, and associations among them and identify the logical sequence of activities undergoing in a system, and represent them pictorially
- Design test cases using Black box and White box testing strategies.
- Use modern CASE tools for designing and testing Software applications.

SE4201

SOFTWARE REQUIREMENTS ENGINEERING

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

COURSE 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 INTRODUCTION

9

Introduction to Requirements – System- Stakeholder- Requirement- Attribute-Vision- Function – Performance –Objective- Quality- Resource Saving –Workload Capacity- Resource- Cost –Budget Design Idea –Condition –Target- Constraint –Benchmark- Introduction to Requirements Engineering-What is Requirements Engineering-What are requirements?-Requirements Engineering activities – Understanding requirements – classification based on functionality considerations, product construction, source-Levels of requirement-Evolution –Ambiguity in requirements specification

UNIT II REQUIREMENTS ELICITATION

9

Requirements Elicitation: Preparing for requirements elicitation. Elicitation techniques survey-eliciting non-functional requirements- prevalence of requirements elicitation techniques-eliciting hazards – Elicitation and gathering of requirements – Process of requirements elicitation – approaches for requirements elicitation methodology based requirements elicitation –tool support – issues –trends and challenges

UNIT III REQUIREMENTS ANALYSIS

9

Identification of Functional and Non Functional Requirements – Identification of Performance Requirements – Six Quality Attributes– Analysis –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 –Requirements engineering for Agile methods- Requirements Engineering for Web-Based Information Systems.

TOTAL: 45 PERIODS

SUGGEST ACTIVITY:

- Students would be asked to identify a problem and frame the problem statement.
- Identify functional/non-functional requirements, domain requirements, and user and system requirements and analyze the feasibility.
- Give a presentation on the work done.

COURSE OUTCOMES

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

CO1:Prepare SRS including the details of requirements engineering

CO2:Describe the stages of requirements elicitation.

CO3:Analyze software requirements gathering.

CO4:Use various methodologies for requirements development.

CO5:Perform requirements validation.

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. Wiegers, Karl, Joy Beatty, "Software requirements", Pearson Education, 2013
5. Aybüke Aurum · Claes Wohlin (Eds.-Engineering and managing software requirements), Springer-Verlag Berlin Heidelberg 2005
6. Julio Cesar Sampaio do Prado Leite and Jorge Haracio Doom-Perspectives on Software Requirements, Springer Science+Business Media New York 2004
7. Phillip A. Laplante – Requirements Engineering for Software and Systems-Auerbach Publications(Applied Software Engineering Series) - (2017)
8. Tom Gilb – Competitive Engineering_ A Handbook For Systems Engineering, Requirements Engineering, and Software Engineering Using Planguage , Elsevier(2005)

COURSE OBJECTIVES

The student should be able to:

- Understand the fundamentals of object modeling.
- Learn the unified process phases.
- Prepare the requirements for various case studies.
- Appreciate the idea behind Design Patterns in handling common problems faced during building an application.
- To practice object modeling using UML

UNIT I INTRODUCTION**9**

Introduction to OOAD; typical activities / workflows / disciplines in OOAD, Introduction to iterative development and the Unified Process, Introduction to UML; mapping disciplines to UML artifacts, Introduction to Design Patterns – goals of a good design, Introducing a case study & MVC architecture.

UNIT II INCEPTION**9**

Artifacts in inception, Understanding requirements – the FURPS model, Understanding Use case model – introduction, use case types and formats, Writing use cases – goals and scope of a use case, elements / sections of a use case, Use case diagrams, Use cases in the UP context and UP artifacts, Identifying additional requirements, Writing requirements for the case study in the use case model.

UNIT III ELABORATION**9**

System sequence diagrams for use case model, Domain model : identifying concepts, adding associations, adding attributes, Interaction Diagrams, Introduction to GRASP design Patterns ,Design Model: Use case realizations with GRASP patterns, Design Class diagrams in each MVC layer Mapping Design to Code, Design class diagrams for case study and skeleton code

UNIT IV DESIGN PATTERNS**9**

Fabrication, Indirection, Singleton, Factory, Facade, Publish-Subscribe

UNIT V UML DIAGRAMS**9**

State-Chart diagrams, Activity diagrams, Component Diagrams, Deployment diagrams, Object diagrams. Advanced concepts in OOAD : Use case relationships, Generalizations Domain Model refinements, Architecture, Packaging model elements.

TOTAL : 45 PERIODS**COURSE OUTCOMES**

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

CO1:Use UML notations Apply UML Use case Notations to applications.

CO2:Can apply unified process in software development

CO3:Understand the best use of Object-Oriented concepts for creating truly OOP programs

CO4:Use design patterns for better class and object composition.

CO5:Understand the concepts of Model refinement and diagrams.

REFERENCES

1. 'Applying UML and patterns' by Craig Larman, Pearson, 2005

2. "Object-Oriented Analysis & Design with the Unified Process" , Satzinger, Jackson & Burd Cengage Learning, Cengage Learning India Publisher,Year 2007
3. 'UML distilled' by Martin Fowler, Addison Wesley,Third Edition 2003
4. O'Reilly ' s 'Head-First Design Patterns' by Eric Freeman et al.Year 2004
5. UML2 Toolkit by Hans-Erik Eriksson, Magnus Penker, Brian Lyons, David Fado:Wiley India Edition, Year 2003

SE4203

SOFTWARE TESTING

L T P C
3 0 0 3

COURSE OBJECTIVES

The student should be able to

- Understand the basics of software testing
- Appreciate the different aspects of testing techniques
- Understand the testing process management
- Know the testing tools and test automation
- Learn the testing of various applications

UNIT I INTRODUCTION

8

Introduction to software testing – Evolution of Software Testing – Goals of Software Testing - Software Testing Definitions - Model for Software Testing - Software Testing as a Process - software testing terminology and methodology – Software Testing Terminology – STLC – Software Testing Methodology – Verification and Validation

UNIT II TESTING TECHNIQUES

10

Black-Box Testing Techniques – Equivalence Class Testing - State Table-Based Testing - Decision Table-Based Testing – Cause-Effect Graphing Based Testing – Error Guessing – White-Box Testing Techniques - Need – Basis Path Testing – Graph Matrices – Loop Testing – Data Flow Testing – Mutation Testing – Static Testing – Progressive vs. Regressive Testing – Regression Testing Techniques

UNIT III MANAGING THE TESTING PROCESS

9

Test Management – Software Metrics – Testing Metrics for Monitoring and Controlling the Testing Process – Efficient Test Suit Management

UNIT IV BUILDING AGILITY & TOOL SUPPORT

9

Building Agility into the Testing Process- Using Agile Methods to Improve Software Testing — Tool Support For Testing

UNIT V TEST AUTOMATION & TESTING THE APPLICATIONS

9

Test Automation – Software Testability – Components of a test case – A test Automation framework -Testing Object-Oriented Software – Testing Web-Based Systems – Testing Web Applications – Functional Web testing with Twill – Selenium – Testing a simple Web Application – Testing Mobile Smartphone Applications

TOTAL : 45 PERIODS

Activity:

Develop small applications; create test scenarios and carry out different types of testing.

COURSE OUTCOMES:

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

CO1:Comprehend a range of testing techniques

CO2:Select an appropriate testing strategy

CO3:Manage the testing process

CO4:Use different tools for testing

CO5:Understand automation testing and test various applications

REFERENCES:

1. Naresh Chauhan, Principles and Practices, Oxford University Press 2010.
2. William Perry, "Effective Methods for Software Testing", John Wiley, 2009
3. C. Titus Brown, Gheorghe Gheorghiu, Jason Huggins, "An Introduction to Testing Web Applications with twill and Selenium", O'Reilly Media, Inc., 2007
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. Brian Hambling, Software Testing An ISTQB–ISEB Foundation Guide Second Edition, 2010
6. Paul Ammann, Jeff Offutt, Introduction to Software Testing, Second Edition, Cambridge University Press, 2017.

SE4204

INTEGRATED SOFTWARE PROJECT MANAGEMENT

L T P C
3 0 0 3

COURSE 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 – PERTEExecution 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

Activity:

A mini-project can be given to the students and use it as a context for the tutorials

COURSE OUTCOMES

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

CO1:Identify the various elements of software management process framework

CO2:Use available open source estimation tools for cost estimation

CO3:Identify existing risk and perform risk assessment

CO4:Design a software metric for software project management

CO5:Learn and assess the practices of people management

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", 3rd edition, CRC Press, 2015.
4. Stark, John, "Decision Engineering: Product Lifecycle Management: 21st Century Paradigm for Product Realisation ", 2nd Edition., Springer London, 2011

SE4211

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

Activity	Instructions	Submission week	Evaluation
Selection of area of interest and Topic Stating an Objective	You are requested to select an area of interest, topic and state an objective	2 nd week	3 % Based on clarity of thought, current relevance and clarity in writing
Collecting Information about your area & topic	<ol style="list-style-type: none"> 1. List 1 Special Interest Groups or professional society 2. List 2 journals 3. List 2 conferences, symposia or workshops 4. List 1 thesis title 5. List 3 web presences (mailing lists, forums, news sites) 6. List 3 authors who publish regularly in your area 7. Attach a call for papers (CFP) from your area. 	3 rd week	3% (the selected information must be area specific and of international and national standard)
Collection of Journal papers in the topic in the context of the objective – collect 20 & then filter	<ul style="list-style-type: none"> • You have to provide a complete list of references you will be using- Based on your objective -Search various digital libraries and Google Scholar • When picking papers to read - try to: <ul style="list-style-type: none"> • 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, • Favour papers from well-known journals and conferences, • Favour “first” or “foundational” papers in the field (as indicated in other people’s survey paper), • Favour more recent papers, • Pick a recent survey of the field so you can quickly gain an overview, 	4 th week	6% (the list of standard papers and reason for selection)

	<ul style="list-style-type: none"> • Find relationships with respect to each other and to your topic area (classification scheme/categorization) • Mark in the hard copy of papers whether complete work or section/sections of the paper are being considered 		
Reading and notes for first 5 papers	<p>Reading Paper Process</p> <ul style="list-style-type: none"> • For each paper form a Table answering the following questions: • What is the main topic of the article? • What was/were the main issue(s) the author said they want to discuss? • Why did the author claim it was important? • How does the work build on other's work, in the author's opinion? • What simplifying assumptions does the author claim to be making? • What did the author do? • How did the author claim they were going to evaluate their work and compare it to others? • What did the author say were the limitations of their research? • What did the author say were the important directions for future research? <p>Conclude with limitations/issues not addressed by the paper (from the perspective of your survey)</p>	5 th week	8% (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 next5 papers	Repeat Reading Paper Process	6 th week	8% (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 th week	8% (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 th week	8% (this component will be evaluated based on the linking and classification among the papers)
Abstract	Prepare a draft abstract and give a presentation	9 th week	6% (Clarity, purpose and conclusion) 6% Presentation & Viva Voce
Introduction Background	Write an introduction and background sections	10 th 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 th 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 th week	5% (conclusions – clarity and your ideas)
Final Draft	Complete the final draft of your paper	13 th week	10% (formatting, English, Clarity and linking) 4% Plagiarism Check Report
Seminar	A brief 15 slides on your paper	14 th & 15 th week	10% (based on presentation and Viva-voce)

TOTAL: 30 PERIODS

SE4212

SOFTWARE DEVELOPMENT LABORATORY

L T P C

0 0 4 2

COURSE OBJECTIVES:

The student should be able to:

- learn the stages of software development
- know about preparing software project documentation
- learn various testing mechanisms
- gain practical experience in applying agile methodology

- understand the principles of DevOps

LIST OF EXERCISES:

Choose any application and apply the phases of Software Development Life Cycle

1. Project Planning

Thorough study of the problem by reviewing the literature – Identify project scope, Objectives, Infrastructure. – PROJECT PLAN DOCUMENTATION

2. Software requirement Analysis

Classify the functional and non-functional requirements - Describe the individual Phases / Modules of the project, Identify deliverables. – SRS DOCUMENTATION

3. Software Design/Modeling

Prepare high-level and low-level designs

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 and Site monitor. – TEST CASE DOCUMENTATION

6. Develop any software application using agile method.

7. Develop any software application using DevOps.

SUGGESTED LIST OF APPLICATIONS:

1. Student Marks Analyzing System.
2. Quiz System.
3. Online Ticket Reservation System
4. Payroll System
5. Course Registration System
6. Stock Maintenance.
7. Book Lending Application.
8. Inventory system
9. Online Payment system
10. Hotel management system

TOTAL:60 PERIODS

COURSE OUTCOMES:

At the end the student will be able to:

CO1:Formulate project plan and SRS

CO2:Prepare design and code documents at appropriate stages of software development

CO3:Test the software product

CO4:Develop a flexible software product using agile.

CO5:Implement DevOps principles to produce high-quality software

COURSE OBJECTIVES

The student should be able to

- Introduce the basics of software reliability
- Understand the various reliable modeling techniques
- Explore the different software reliable models
- Test the product for quality
- Monitor and comply against the defined standards

UNIT I INTRODUCTION TO SOFTWARE RELIABILITY 9

Defining failure – choosing a common measure – System and software failure intensity objectives – software reliability strategies - Failures, Faults and Errors – Availability – system and component reliabilities – basic failure intensity - Need for reliable software – concepts - The Dependability Concept - Failure Behavior of an X- ware System

UNIT II SOFTWARE RELIABILITY MODELING 9

Software Reliability Modeling Survey - Concepts – Common Distribution Function – Generalised Systemability Function – System Reliability – Markov Processes – Counting Processes

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 SOFTWARE QUALITY ASSURANCE 9

Software Quality - Quality Principles - Quality Factors: Product operation, revision and transition; Components of SQA: System and architecture; Pre-Project Components; Contract Review; Development and Quality Plans; SQA Components in Project Life Cycle: SQA defect removal policies; Reviews; Project progress control; Costs; Quality Management Standards; Project Process Standards; Management and its Role in SQA; SQA Unit.

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 PERIODS

Activities:

Give a presentation about a software reliability tool of students' choice.

Create an SQA Management Plan.

COURSE OUTCOMES:

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

CO1: Perform some simple statistical analysis relevant to software measurement data

CO2: Compare and pick out the right reliability model

CO3: Evaluate the reliability of any given software product

CO4: Develop Quality plans and use SQA components in project life cycle

CO5: Assess Quality standards of various software products

REFERENCES:

1. John D. Musa, "Software Reliability Engineering", Tata McGraw Hill, 1999
2. Michael R. Lyu - Handbook of software reliability engineering-IEEE Computer Society Press_ McGraw Hill (1996)
3. John D. Musa, Anthony Iannino, KazuhiraOkumoto, "Software Reliability – Measurement, Prediction, Application, Series in Software Engineering and Technology", McGraw Hill, 1987
4. DanielGalín, "Software Quality Assurance – From Theory to Implementation", Pearson Education, 2009.
5. Westfall, Linda - The Certified Software Quality Engineer Handbook-ASQ Quality Press (2009)
6. Hoang Pham, System Software Reliability- (Springer Series in Reliability Engineering Verlag London (2007)
7. John D. Musa - Software Reliability Engineering_ More Reliable Software Faster and Cheaper 2nd Edition-AuthorHouse (2004)

SE4071

AGILE METHODOLOGIES

L T P C

3 0 0 3

COURSE OBJECTIVES:

- To learn the fundamental principles and practices associated with each of the agile development methods
- To apply the principles and practices of agile software development on a project of interest and relevance to the student.
- To provide a good understanding of software design and a set of software technologies and APIs.
- To do a detailed examination and demonstration of Agile development and testing techniques.
- To understand Agile development and testing.

UNIT I AGILE SOFTWARE DEVELOPMENT

9

Basics and Fundamentals of Agile Process Methods, Values of Agile, Principles of Agile, stakeholders, Challenges . Lean Approach: Waste Management, Kaizen and Kanban, add process and products add value. Roles related to the lifecycle, differences between Agile and traditional plans, differences between Agile plans at different lifecycle phases. Testing plan links between testing, roles and key techniques, principles, understand as a means of assessing the initial status of a project/ How Agile helps to build quality

UNIT II AGILE AND SCRUM PRINCIPLES

9

Agile Manifesto, Twelve Practices of XP, Scrum Practices, Applying Scrum. Need of scrum, working of scrum, advanced Scrum Applications, Scrum and the Organization, scrum values

UNIT III AGILE PRODUCT MANAGEMENT

9

Communication, Planning, Estimation Managing the Agile approach Monitoring progress, Targeting and motivating the team, Managing business involvement, Escalating issue. Quality, Risk, Metrics and Measurements, Managing the Agile approach Monitoring progress, Targeting and motivating the team, Managing business involvement and Escalating issue

UNIT IV AGILE REQUIREMENTS AND AGILE TESTING**9**

User Stories, Backlog Management. Agile Architecture: Feature Driven Development. Agile Risk Management: Risk and Quality Assurance, Agile Tools. Agile Testing Techniques, Test-Driven Development, User Acceptance Test

UNIT V AGILE REVIEW AND SCALING AGILE FOR LARGE PROJECTS**9**

Agile Metrics and Measurements, The Agile approach to estimating and project variables, Agile Measurement, Agile Control: the 7 control parameters. Agile approach to Risk, The Agile approach to Configuration Management, The Atern Principles, Atern Philosophy, The rationale for using Atern, Refactoring, Continuous integration, Automated Build Tools. Scrum of Scrums, Team collaborations, Scrum, Estimate a Scrum Project, Track Scrum Projects, Communication in Scrum Projects, Best Practices to Manage Scrum.

COURSE OUTCOMES:

CO1: Analyze existing problems with the team, development process and wider organization

CO2: Apply a thorough understanding of Agile principles and specific practices

CO3: Select the most appropriate way to improve results for a specific circumstance or need

CO4: Judge and craft appropriate adaptations to existing practices or processes depending upon analysis of typical problems

CO5: Evaluate likely successes and formulate plans to manage likely risks or problems

TOTAL: 45 PERIODS**REFERENCES**

1. Robert C. Martin ,Agile Software Development, Principles, Patterns, and Practices Alan Apt Series (2011)
2. Succeeding with Agile : Software Development Using Scrum, Pearson (2010)
3. David J. Anderson and Eli Schragenheim, "Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results, Prentice Hall, 2003.
4. Hazza and Dubinsky, "Agile Software Engineering, Series: Undergraduate Topics in Computer Science, Springer, 2009.
5. Craig Larman, "Agile and Iterative Development: A Managers Guide, Addison-Wesley, 2004.
6. Kevin C. Desouza, "Agile Information Systems: Conceptualization, Construction, and Management, Butterworth-Heinemann, 2007.

IF4095**SOCIAL NETWORK ANALYSIS****L T P C****3 0 0 3****COURSE OBJECTIVES:**

- Formalise different types of entities and relationships as nodes and edges and represent this information as relational data.
- Understand the fundamental concepts in analyzing the large-scale data that are derived from social networks
- Understand the basic concepts and principles of different theoretical models of social networks analysis.
- Transform data for analysis using graph-based and statistics-based social network

measures

- Choose among social network designs based on research goals

UNIT I GRAPH THEORY AND STRUCTURE 10

Breadth First Search (BFS) Algorithm. Strongly Connected Components (SCC) Algorithm. Weakly Connected Components (WCC) Algorithm. First Set of Experiments—Degree Distributions. Second Set of Experiments—Connected Components. Third Set of Experiments—Number of Breadth First Searches. Rank Exponent R. Out-Degree Exponent O. Hop Plot Exponent H. Eigen Exponent E. Permutation Model. Random Graphs with Prescribed Degree Sequences. Switching Algorithms. Matching Algorithm. "Go with the Winners" Algorithm. HyperANF Algorithm. Iterative Fringe Upper Bound (iFUB) Algorithm. Spid. Degree Distribution. Path Length. Component Size. Clustering Coefficient and Degeneracy. Friends-of-Friends. Degree Assortativity. Login Correlation.

UNIT II SOCIAL NETWORK GRAPH ANALYSIS 9

Social network exploration/ processing and properties: Finding overlapping communities, similarity between graph nodes, counting triangles in graphs, neighborhood properties of graphs. Pregel paradigm and Apache Giraph graph processing system.

UNIT III INFORMATION DIFFUSION IN SOCIAL NETWORKS 9

Strategic network formation: game theoretic models for network creation/ user behavior in social networks. Information diffusion in graphs: Cascading behavior, spreading, epidemics, heterogeneous social network mining, influence maximization, outbreak detection. Opinion analysis on social networks: Contagion, opinion formation, coordination and cooperation.

UNIT IV CASCADING IN SOCIAL NETWORKS 8

Cascading in Social Networks. Decision Based Models of Cascade. Collective Action. Cascade Capacity. Co-existence of Behaviours. Cascade Capacity with Bilinguality. Probabilistic Models of Cascade. Branching Process. Basic Reproductive Number. SIR Epidemic Model. SIS Epidemic Model. SIRS Epidemic Model. Transient Contact Network. Cascading in Twitter.

UNIT V LINK ANALYSIS & COMMUNITY DETECTION 9

Search Engine. Crawling. Storage. Indexing. Ranking. Google. Data Structures. Crawling. Searching. Web Spam Pages Strength of Weak Ties. Triadic Closure. Detecting Communities in a Network. Girvan-Newman Algorithm. Modularity. Minimum Cut Trees. Tie Strengths in Mobile Communication Network. Exact Betweenness Centrality. Approximate Betweenness Centrality.

SUGGESTED ACTIVITIES:

- 1: Twitter Intelligence project performs tracking and analysis of the Twitter
- 2: Large-Scale Network Embedding as Sparse Matrix Factorization
- 3: Implement how Information Propagation on Twitter
- 4: Social Network Analysis and Visualization software application.
- 5: Implement the Structure of Links in Networks

COURSE OUTCOMES:

- CO1:** Plan and execute network analytical computations.
- CO2:** Implement mining algorithms for social networks
- CO3:** Analyze and evaluate social communities.
- CO4:** Use social network analysis in behavior analytics
- CO5:** Perform mining on large social networks and illustrate the results.

REFERENCES

1. Practical Social Network Analysis with Python, Krishna Raj P. M. Ankith Mohan and K. G. Srinivasa. Springer, 2018
2. SOCIAL NETWORK ANALYSIS: METHODS AND APPLICATIONS, STANLEY WASSERMAN, and KATHERINE F' AUST. CAMBRIDGE UNIVERSITY PRESS, 2012
3. Social Network Analysis: History, Theory and Methodology by Christina Prell, SAGE Publications, 1st edition, 2011
4. Sentiment Analysis in Social Networks, Federico Alberto Pozzi, Elisabetta Fersini, Enza Messina, and Bing. LiuElsevier Inc, 1st edition, 2016
5. Social Network Analysis, John Scott. SAGE Publications, 2012

BC4152

CYBER FORENSICS AND INVESTIGATION

L T P C

3 0 0 3

COURSE OBJECTIVES:

- To gain a comprehensive understanding of cyber forensic principles and the collection, preservation, and analysis of digital evidence
- To combine both the technical expertise and the knowledge required to investigate, detect and prevent digital crimes.
- To understand the different applications and methods for conducting network and digital forensic acquisition and analysis
- To learn the E-evidence collection and preservation, investigating operating systems and file systems, network, cloud and mobile device forensics
- To gain knowledge on digital forensics legislations, digital crime, forensic processes and procedures.

UNIT I CYBER FORENSICS SCIENCE

9

Cyber Forensics Science: Forensics Science, Forensics Fundamentals, Computer Forensics, and Digital Forensics.

Cyber Crime: Criminalistics as it relates to the Investigative Process, Analysis of Cyber Criminalistics Area, Holistic Approach to Cyber-forensics, Computer Forensics and Law Enforcement- Indian Cyber Forensic - Forensics Services, Professional Forensics Methodology- Types of Forensics Technology

UNIT II NETWORK SECURITY FORENSICS SYSTEM AND SERVICES

9

Forensics system and Services : Forensics on - Internet Usage – Intrusion - Firewall and Storage Area Network; Occurrence of Cyber-crimes- Cyber Detectives- Fighting Cyber Crimes- Forensic Process

Open-source Security Tools for Network Forensic Analysis, Requirements for Preservation of Network Data

Computer Forensics - Data Backup and Recovery - Test Disk Suite.

UNIT III DIGITAL FORENSICS PRESERVATION AND FORENSIC DATA ANALYSIS

9

Digital Repositories - Evidence Collection – Data Preservation Approaches – Meta Data and Historic records – Legal aspects. Basic Steps of Forensic Analysis in Windows and Linux –

Forensic Scenario – Email Analysis – File Signature Analysis – Hash Analysis – Forensic Examination of log files
Data-Recovery Solution, Hiding and Recovering Hidden Data, Evidence Collection and Data Seizure

UNIT IV CLOUD, NETWORK AND MOBILE FORENSICS 9

Working with the cloud vendor, obtaining evidence, reviewing logs and APIs
Mobile Forensics techniques, Mobile Forensics Tools - Android Device – Analysis- Android Malware – iOS Forensic Analysis – SIM Forensic Analysis – Case study
Recent trends in Mobile Forensic Technique and methods to Search and Seize Electronic Evidence

UNIT V LEGAL ASPECTS OF DIGITAL FORENSICS 9

IT Laws and Ethics, Digital Evidence Controls, Evidence Handling Procedures, Basics of Indian Evidence ACT IPC and CrPC , Electronic Communication Privacy ACT, Legal Policies, Act 2000, amendment of IT Act 2008.

Current Cyber Forensic Tools: Overview of different software packages – Encase-Autopsy-Magnet – Wireshark - Mobile Forensic Tools – SQLite

TOTAL : 45 PERIODS

COURSE OUTCOMES:

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

CO1: Understand the responsibilities and liabilities of a computer forensic investigator

CO2: Identify potential sources of electronic evidence.

CO3: Understand the importance of maintaining the integrity of digital evidence.

CO4: Demonstrate the ability to perform basic forensic data acquisition and analysis using computer and network based applications and utilities.

CO5: Understand relevant legislation and codes of ethics.

REFERENCES:

1. J. R. Vacca, Computer forensics: Computer Crime Scene investigation, 2nd Ed. Hanover, NH, United States: Charles River Media, 2002, Laxmi Publications, 1st Edition, 2015.
2. C. Altheide, H. Carvey, and R. Davidson, Digital Forensics with Open Source Tools: Using Open Source Platform Tools for Performing Computer Forensics on Target Systems: Windows, Mac, Linux, Unix, etc, 1st Ed. United States: Syngress, 2011.
3. S. Bommisetty, R. Tamma, and H. Mahalik, Practical Mobile Forensics: Dive into Mobile Forensics on IOS, Android, Windows, and blackBerry devices with this action-packed, practical guide. United Kingdom: Packt Publishing, 2014.
4. G. Gogolin, Digital Forensics Explained, 1st Ed. Boca Raton, FL: CRC Taylor & Francis, 1st Edition, Auerbach Publications, 2013.
5. A. Hoog and J. McCash, Android forensics: Investigation, Analysis, and Mobile Security for Google Android. Waltham, MA: Syngress Media, U.S., 2011.
6. B. Nelson, A. Phillips, F. Enfinger, and C. Steuart, Guide to Computer Forensics and Investigations, Second edition, 2nd Ed. Boston: Thomson Course Technology, 2009.
7. C. Altheide and H. Carvey, "Digital Forensics with Open Source Tools", 2011 Publisher(s): Syngress.
8. J. Sammons, "The Basics of Digital Forensics- The Primer for Getting Started in Digital Forensics", 1st Edition, Syngress, 2012.
9. Nelson, Phillips and Enfinger Steuart, "Guide to Computer Forensics and Investigations", 6th Edition, Cengage Learning, New Delhi, 2020.

COURSE OBJECTIVES:

- To gain expertise in Virtualization, Virtual Machines and deploy practical virtualization solution
- To understand the architecture, infrastructure and delivery models of cloud computing.
- To explore the roster of AWS services and illustrate the way to make applications in AWS
- To gain knowledge in the working of Windows Azure and Storage services offered by Windows Azure
- To develop the cloud application using various programming model of Hadoop and Aneka

UNIT I VIRTUALIZATION AND VIRTUALIZATION INFRASTRUCTURE 6

Basics of Virtual Machines - Process Virtual Machines – System Virtual Machines –Emulation – Interpretation – Binary Translation - Taxonomy of Virtual Machines. Virtualization –Management Virtualization — Hardware Maximization – Architectures – Virtualization Management – Storage Virtualization – Network Virtualization- Implementation levels of virtualization – virtualization structure – virtualization of CPU, Memory and I/O devices – virtual clusters and Resource Management – Virtualization for data center automation

UNIT II CLOUD PLATFORM ARCHITECTURE 12

Cloud Computing: Definition, Characteristics - Cloud deployment models: public, private, hybrid, community – Categories of cloud computing: Everything as a service: Infrastructure, platform, software- A Generic Cloud Architecture Design – Layered cloud Architectural Development – Architectural Design Challenges

UNIT III AWS CLOUD PLATFORM - IAAS 9

Amazon Web Services: AWS Infrastructure- AWS API- AWS Management Console - Setting up AWS Storage - Stretching out with Elastic Compute Cloud - Elastic Container Service for Kubernetes- AWS Developer Tools: AWS Code Commit, AWS Code Build, AWS Code Deploy, AWS Code Pipeline, AWS code Star - AWS Management Tools: Cloud Watch, AWS Auto Scaling, AWS control Tower, Cloud Formation, Cloud Trail, AWS License Manager

UNIT IV PAAS CLOUD PLATFORM 9

Windows Azure: Origin of Windows Azure, Features, The Fabric Controller – First Cloud APP in Windows Azure- Service Model and Managing Services: Definition and Configuration, Service runtime API- Windows Azure Developer Portal- Service Management API- Windows Azure Storage Characteristics-Storage Services- REST API- Blops

UNIT V PROGRAMMING MODEL 9

Introduction to Hadoop Framework - Mapreduce, Input splitting, map and reduce functions, specifying input and output parameters, configuring and running a job –Developing Map Reduce Applications - Design of Hadoop file system –Setting up Hadoop Cluster- Aneka: Cloud Application Platform, Thread Programming, Task Programming and Map-Reduce Programming in Aneka

COURSE OUTCOMES:

- CO1:** Employ the concepts of virtualization in the cloud computing
CO2: Identify the architecture, infrastructure and delivery models of cloud computing
CO3: Develop the Cloud Application in AWS platform
CO4: Apply the concepts of Windows Azure to design Cloud Application

CO5: Develop services using various Cloud computing programming models.

TOTAL : 45 PERIODS

REFERENCES

1. Bernard Golden, Amazon Web Service for Dummies, John Wiley & Sons, 2013.
2. Raoul Alongi, AWS: The Most Complete Guide to Amazon Web Service from Beginner to Advanced Level, Amazon Asia- Pacific Holdings Private Limited, 2019.
3. Sriram Krishnan, Programming: Windows Azure, O'Reilly, 2010.
4. Rajkumar Buyya, Christian Vacchiola, S.Thamarai Selvi, Mastering Cloud Computing , MCGraw Hill Education (India) Pvt. Ltd., 2013.
5. Danielle Ruest, Nelson Ruest, "Virtualization: A Beginner's Guide", McGraw-Hill Osborne Media, 2009.
6. Jim Smith, Ravi Nair , "Virtual Machines: Versatile Platforms for Systems and Processes", Elsevier/Morgan Kaufmann, 2005.
7. John W.Rittinghouse and James F.Ransome, "Cloud Computing: Implementation, Management, and Security", CRC Press, 2010.
8. Toby Velte, Anthony Velte, Robert Elsenpeter, "Cloud Computing, A Practical Approach", McGraw-Hill Osborne Media, 2009.
9. Tom White, "Hadoop: The Definitive Guide", Yahoo Press, 2012.

SE4072

IMAGE PROCESSING

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

COURSE OBJECTIVES:

- To study fundamental concepts of digital image processing.
- To get exposed to simple image enhancement techniques in Spatial and Frequency domain.
- To become familiar with image compression
- To study the image segmentation and Morphological Processing.
- To expose student's in recognition methods.

UNIT I INTRODUCTION

9

Examples of fields that use digital image processing, fundamental steps in digital image processing, components of image processing system. Digital Image Fundamentals: A simple image formation model, image sampling and quantization, basic relationships between pixels. Color Image Processing: Color fundamentals, color models, pseudo color image processing, basics of full-color image processing, color transforms, smoothing and sharpening, color segmentation

UNIT II IMAGE ENHANCEMENT

9

Image enhancement in the spatial domain: Basic gray-level transformation, histogram processing, enhancement using arithmetic and logic operators, basic spatial filtering, smoothing, and sharpening spatial filters, combining the spatial enhancement methods. Filtering in the Frequency Domain: Preliminary Concepts, Extension to functions of two variables, Image Smoothing, Image Sharpening, Homomorphic filtering. A model of the image degradation/restoration process, noise models, restoration in the presence of noise-only spatial filtering,

UNIT III WAVELETS AND IMAGE COMPRESSION

9

Wavelets and Multiresolution Processing. Fundamentals, image compression models, error-free

compression, lossy predictive coding, image compression standards

UNIT IV IMAGE SEGMENTATION

9

Detection of Discontinuities, Edge Linking and Boundary Detection, Thresholding, Region-Based Segmentation, Segmentation by Morphological Watersheds, The Use of Motion in Segmentation Morphological Image Processing: Preliminaries, dilation, erosion, open and closing, hit or miss transformation, basic morphologic algorithms.

UNIT V REPRESENTATION AND OBJECT RECOGNITION

9

Representation, Boundary Descriptors, Regional Descriptors, Use of Principal Components for Description. Object Recognition: Patterns and patterns classes, recognition based on decision-theoretic methods, matching, optimum statistical classifiers, neural networks, structural methods – matching shape numbers, string matching.

COURSE OUTCOMES:

- CO1:** Apply knowledge of mathematics for image understanding and analysis.
- CO2:** Design and analysis of techniques / processes for image Enhancement.
- CO3:** Design and analysis of techniques / processes for image compression.
- CO4:** Able to expose to current trends in field of image segmentation.
- CO5:** Design, realize and troubleshoot various algorithms for image processing case studies.

TOTAL : 45 PERIODS

REFERENCES

1. Digital Image Processing, Rafeal C.Gonzalez, Richard E.Woods, fourth Edition, Pearson Education/PHI, 2018
2. Image Processing, Analysis, and Machine Vision, Milan Sonka, Vaclav Hlavac and Roger Boyle, fourth Edition, Thomson Learning, 2015
3. Introduction to Digital Image Processing with Matlab, Alasdair McAndrew, Thomson Course Technology, 2021
4. Computer Vision and Image Processing, Adrian Low, Second Edition, B.S.Publications,2022
5. Digital Image Processing using Matlab, Rafeal C.Gonzalez, Richard E.Woods, Steven L. Eddins, Pearson Education,2006.

CP4093

INFORMATION RETRIEVAL TECHNIQUES

L T P C

3 0 0 3

COURSE 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 get an understanding of machine learning techniques for text classification and clustering.
- 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 EVALUATION AND PARALLEL INFORMATION RETRIEVAL 9

Traditional Effectiveness Measures – Statistics in Evaluation – Minimizing Adjudication Effect – Nontraditional Effectiveness Measures – Measuring Efficiency – Efficiency Criteria –Queueing Theory – Query Scheduling – Parallel Information Retrieval – Parallel Query Processing – MapReduce

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

COURSE OUTCOMES:

CO1: Build an Information Retrieval system using the available tools.

CO2: Identify and design the various components of an Information Retrieval system.

CO3: Categorize the different types of IR Models.

CO4: Apply machine learning techniques to text classification and clustering which is used for efficient Information Retrieval.

CO5: 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. Stefan Butcher, Implementing and Evaluating Search Engines, The MIT Press, Cambridge, Massachusetts London, England, 2016.
3. Ricardo Baeza – Yates, Berthier Ribeiro – Neto, "Modern Information Retrieval: The concepts and Technology behind Search (ACM Press Books), Second Edition, 2011.
4. Stefan Butcher, Charles L. A. Clarke, Gordon V. Cormack, "Information Retrieval

COURSE OBJECTIVES:

- To familiarize Use the Innovation Canvas to justify potentially successful products.
- To learn various ways in which to develop a product idea.
- To understand about how Big Data can play vital role in Cognitive Computing
- To know about the business applications of Cognitive Computing
- To get into all applications of Cognitive Computing

UNIT I FOUNDATION OF COGNITIVE COMPUTING 9

Foundation of Cognitive Computing: cognitive computing as a new generation, the uses of cognitive systems, system cognitive, gaining insights from data, Artificial Intelligence as the foundation of cognitive computing, understanding cognition Design Principles for Cognitive Systems: Components of a cognitive system, building the corpus, bringing data into cognitive system, machine learning, hypotheses generation and scoring, presentation, and visualization services

UNIT II NATURAL LANGUAGE PROCESSING IN COGNITIVE SYSTEMS 9

Natural Language Processing in support of a Cognitive System: Role of NLP in a cognitive system, semantic web, Applying Natural language technologies to Business problems Representing knowledge in Taxonomies and Ontologies: Representing knowledge, Defining Taxonomies and Ontologies, knowledge representation, models for knowledge representation, implementation considerations

UNIT III BIG DATA AND COGNITIVE COMPUTING 9

Relationship between Big Data and Cognitive Computing: Dealing with human-generated data, defining big data, architectural foundation, analytical data warehouses, Hadoop, data in motion and streaming data, integration of big data with traditional data Applying Advanced Analytics to cognitive computing: Advanced analytics is on a path to cognitive computing, Key capabilities in advanced analytics, using advanced analytics to create value, Impact of open source tools on advanced analytics

UNIT IV BUSINESS IMPLICATIONS OF COGNITIVE COMPUTING 9

Preparing for change ,advantages of new disruptive models , knowledge meaning to business, difference with a cognitive systems approach , meshing data together differently, using business knowledge to plan for the future , answering business questions in new ways , building business specific solutions , making cognitive computing a reality , cognitive application changing the market The process of building a cognitive application: Emerging cognitive platform, defining the objective, defining the domain, understanding the intended users and their attributes, questions and exploring insights, training and testing

UNIT V APPLICATION OF COGNITIVE COMPUTING 9

Building a cognitive health care application: Foundations of cognitive computing for healthcare, constituents in healthcare ecosystem, learning from patterns in healthcare Data, Building on a foundation of big data analytics, cognitive applications across the health care eco system, starting with a cognitive application for healthcare, using cognitive applications to improve health and wellness, using a cognitive application to enhance the electronic medical record Using cognitive

application to improve clinical teaching

COURSE OUTCOMES:

- CO1:** Explain applications in Cognitive Computing.
- CO2:** Describe Natural language processor role in Cognitive computing.
- CO3:** Explain future directions of Cognitive Computing
- CO4:** Evaluate the process of taking a product to market
- CO5:** Comprehend the applications involved in this domain.

TOTAL :45 PERIODS

REFERENCES

1. Judith H Hurwitz, Marcia Kaufman, Adrian Bowles, "Cognitive computing and Big Data Analytics", Wiley, 2015
2. Robert A. Wilson, Frank C. Keil, "The MIT Encyclopedia of the Cognitive Sciences", The MIT Press, 1999.
3. Noah D. Goodman, Joshua B. Tenenbaum, The ProbMods Contributors, "Probabilistic Models of Cognition", Second Edition, 2016, <https://probmods.org/>.

IF4094

PATTERN RECOGNITION

L T P C
3 0 0 3

COURSE OBJECTIVES:

- Understand the in-depth concept of Pattern Recognition
- Implement Bayes Decision Theory
- Understand the in-depth concept of Perception and related Concepts
- Understand the concept of ML Pattern Classification
- Understand the concept of DL Pattern Recognition

UNIT I

PATTERN RECOGNITION

8

Induction Algorithms. Rule Induction. Decision Trees. Bayesian Methods. Overview. Naive Bayes. The Basic Naïve Bayes Classifier. Naive Bayes Induction for Numeric Attributes. Correction to the Probability Estimation. Laplace Correction. No Match. Other Bayesian Methods. Other Induction Methods. Neural Networks. Genetic Algorithms. Instance-based Learning. Support Vector Machines.

UNIT II

STATISTICAL PATTERN RECOGNITION

8

About Statistical Pattern Recognition. Classification and regression. Features, Feature Vectors, and Classifiers. Pre-processing and feature extraction. The curse of dimensionality. Polynomial curve fitting. Model complexity. Multivariate non-linear functions. Bayes' theorem. Decision boundaries. Parametric methods. Sequential parameter estimation. Linear discriminant functions. Fisher's linear discriminant. Feed-forward network mappings.

UNIT III

BAYES DECISION THEORY CLASSIFIERS

11

Bayes Decision Theory. Discriminant Functions and Decision Surfaces. The Gaussian Probability Density Function. The Bayesian Classifier for Normally Distributed Classes. Exact interpolation. Radial basis function networks. Network training. Regularization theory. Noisy interpolation theory. Relation to kernel regression. Radial basis function networks for classification. Comparison with the multi-layer perceptron. Basis function optimization.

UNIT IV LINEAR DISCRIMINANT FUNCTIONS**9**

Linear Discriminant Functions and Decision Surfaces. The Two-Category Case. The Multicategory Case. The Perceptron Criterion Function. Batch Perceptron. Perceptron Algorithm Convergence. The Pocket Algorithm. Mean Square Error Estimation. Stochastic Approximation and the LMS Algorithm. Convergence Proof for Single-Sample Correction. Fixed increment descent. Some Direct Generalizations. Fixed increment descent. Batch variable increment Perceptron. Balanced Winnow algorithm. Relaxation Procedures. The Descent Algorithm.

UNIT V NONLINEAR CLASSIFIERS**9**

The Two Layer Perception. The Three Layer Perception. Algorithms Based On Exact Classification Of The Training Set. Feedforward operation and classification. General feedforward operation. Expressive power of multilayer networks. Backpropagation algorithm. Network learning. Training protocols. Stochastic Backpropagation. Batch Backpropagation. Radial basis function networks (RBF). Special bases. Time delay neural networks (TDNN). Recurrent networks. Counter propagation. Cascade-Correlation. Cascade-correlation. Neocognitron

SUGGESTED ACTIVITIES:

- 1: Car Sales Pattern Classification using Support Vector Classifier
- 2: Avocado Sales Pattern Recognition using Linear regression
- 3: Tracking Movements by implementing Pattern Recognition
- 4: Detecting Lanes by implementing Pattern Recognition
- 5: Pattern Detection in SAR Images

TOTAL:45 PERIODS**COURSE OUTCOMES:**

- CO1:** Discover imaging, and interpretation of temporal patterns
CO2: Identify Structural Data Patterns
CO3: Implement Pattern Classification using Machine Learning Classifiers
CO4: Implement Pattern Recognition using Deep Learning Models
CO5: Implement Image Pattern Recognition

REFERENCES

1. Pattern Classification, 2nd Edition, Richard O. Duda, Peter E. Hart, and David G. Stork. Wiley, 2000
2. Pattern Recognition, Jürgen Beyerer, Matthias Richter, and Matthias Nagel. 2018
3. Pattern Recognition and Machine Learning, Christopher M. Bishop. Springer, 2010
4. Pattern Recognition and Classification, Dougherty, and Geoff. Springer, 2013
5. Practical Machine Learning and Image Processing, Himanshu Singh. Apress, 2019

BD4251**BIG DATA MINING AND ANALYTICS****L T P C
3 0 0 3****COURSE OBJECTIVES:**

- To understand the computational approaches to Modeling, Feature Extraction
- To understand the need and application of Map Reduce
- To understand the various search algorithms applicable to Big Data
- To analyse and interpret streaming data

- To learn how to handle large data sets in main memory and learn the various clustering techniques applicable to Big Data

UNIT I DATA MINING AND LARGE SCALE FILES 9

Introduction to Statistical modeling – Machine Learning – Computational approaches to modeling – Summarization – Feature Extraction – Statistical Limits on Data Mining - Distributed File Systems – Map-reduce – Algorithms using Map Reduce – Efficiency of Cluster Computing Techniques.

UNIT II SIMILAR ITEMS 9

Nearest Neighbor Search – Shingling of Documents – Similarity preserving summaries – Locality sensitive hashing for documents – Distance Measures – Theory of Locality Sensitive Functions – LSH Families – Methods for High Degree of Similarities.

UNIT III MINING DATA STREAMS 9

Stream Data Model – Sampling Data in the Stream – Filtering Streams – Counting Distance Elements in a Stream – Estimating Moments – Counting Ones in Window – Decaying Windows.

UNITIV LINK ANALYSIS AND FREQUENT ITEMSETS 9

Page Rank –Efficient Computation - Topic Sensitive Page Rank – Link Spam – Market Basket Model – A-priori algorithm – Handling Larger Datasets in Main Memory – Limited Pass Algorithm – Counting Frequent Item sets.

UNIT V CLUSTERING 9

Introduction to Clustering Techniques – Hierarchical Clustering –Algorithms – K-Means – CURE – Clustering in Non – Euclidean Spaces – Streams and Parallelism – Case Study: Advertising on the Web – Recommendation Systems.

TOTAL: 45 PERIODS

COURSE OUTCOMES:

Upon completion of this course, the students will be able to

CO1:Design algorithms by employing Map Reduce technique for solving Big Data problems.

CO2:Design algorithms for Big Data by deciding on the apt Features set .

CO3:Design algorithms for handling petabytes of datasets

CO4:Design algorithms and propose solutions for Big Data by optimizing main memory consumption

CO5:Design solutions for problems in Big Data by suggesting appropriate clustering techniques.

REFERENCES:

1. Jure Leskovec, AnandRajaraman, Jeffrey David Ullman, “Mining of Massive Datasets”, Cambridge University Press, 3rd Edition, 2020.
2. Jiawei Han, MichelineKamber, Jian Pei, “Data Mining Concepts and Techniques”, Morgan Kaufman Publications, Third Edition, 2012.
3. Ian H.Witten, Eibe Frank “Data Mining – Practical Machine Learning Tools and Techniques”, Morgan Kaufman Publications, Third Edition, 2011.
4. David Hand, HeikkiMannila and Padhraic Smyth, “Principles of Data Mining”, MIT PRESS, 2001

WEB REFERENCES:

1. https://swayam.gov.in/nd2_arp19_ap60/preview
2. https://nptel.ac.in/content/storage2/nptel_data3/html/mhrd/ict/text/106104189/lec1.pdf

ONLINE RESOURCES:

1. <https://examupdates.in/big-data-analytics/>
2. https://www.tutorialspoint.com/big_data_analytics/index.htm
3. https://www.tutorialspoint.com/data_mining/index.htm

BD4151**FOUNDATIONS OF DATA SCIENCE****L T P C**
3 0 0 3**COURSE OBJECTIVES:**

- To apply fundamental algorithms to process data.
- Learn to apply hypotheses and data into actionable predictions.
- Document and transfer the results and effectively communicate the findings using visualization techniques.
- To learn statistical methods and machine learning algorithms required for Data Science.
- To develop the fundamental knowledge and understand concepts to become a data science professional.

UNIT I INTRODUCTION TO DATA SCIENCE 9

Data science process – roles, stages in data science project – working with data from files – working with relational databases – exploring data – managing data – cleaning and sampling for modeling and validation – introduction to NoSQL.

UNIT II MODELING METHODS 9

Choosing and evaluating models – mapping problems to machine learning, evaluating clustering models, validating models – cluster analysis – K-means algorithm, Naïve Bayes – Memorization Methods – Linear and logistic regression – unsupervised methods.

UNIT III INTRODUCTION TO R 9

Reading and getting data into R – ordered and unordered factors – arrays and matrices – lists and data frames – reading data from files – probability distributions – statistical models in R - manipulating objects – data distribution.

UNIT IV MAP REDUCE 9

Introduction – distributed file system – algorithms using map reduce, Matrix-Vector Multiplication by Map Reduce – Hadoop - Understanding the Map Reduce architecture - Writing Hadoop MapReduce Programs - Loading data into HDFS - Executing the Map phase - Shuffling and sorting - Reducing phase execution.

UNIT V DATA VISUALIZATION 9

Documentation and deployment – producing effective presentations – Introduction to graphical analysis – plot() function – displaying multivariate data – matrix plots – multiple plots in one window - exporting graph using graphics parameters - Case studies.

TOTAL : 45 PERIODS**COURSE OUTCOMES:**

- CO1:** Obtain, clean/process and transform data.
- CO2:** Analyze and interpret data using an ethically responsible approach.
- CO3:** Use appropriate models of analysis, assess the quality of input, derive insight from results, and investigate potential issues.

CO4: Apply computing theory, languages and algorithms, as well as mathematical and statistical models, and the principles of optimization to appropriately formulate and use data analyses.

CO5: Formulate and use appropriate models of data analysis to solve business-related challenges.

REFERENCES

1. Nina Zumel, John Mount, "Practical Data Science with R", Manning Publications, 2014.
2. Mark Gardener, "Beginning R - The Statistical Programming Language", John Wiley & Sons, Inc., 2012.
3. W. N. Venables, D. M. Smith and the R Core Team, "An Introduction to R", 2013.
4. Tony Ojeda, Sean Patrick Murphy, Benjamin Bengfort, Abhijit Dasgupta, "Practical Data Science Cookbook", Packt Publishing Ltd., 2014.
5. Nathan Yau, "Visualize This: The FlowingData Guide to Design, Visualization, and Statistics", Wiley, 2011.
7. Boris Lublinsky, Kevin T. Smith, Alexey Yakubovich, "Professional Hadoop Solutions", John Wiley & Sons Inc., 2013.

IF4072

DESIGN THINKING

L T P C
3 0 0 3

COURSE OBJECTIVES:

- To provide a sound knowledge in UI & UX
- To understand the need for UI and UX
- Research Methods used in Design
- Tools used in UI & UX
- Creating a wireframe and prototype

UNIT I UX LIFECYCLE TEMPLATE

8

Introduction. A UX process lifecycle template. Choosing a process instance for your project. The system complexity space. Meet the user interface team. Scope of UX presence within the team. More about UX lifecycles. Business Strategy. Value Innovation. Validated User Research. Killer UX Design. The Blockbuster Value Proposition. What Is a Value Proposition?.

UNIT II CONTEXTUAL INQUIRY

10

The system concept statement. User work activity data gathering. Look for emotional aspects of work practice. Abridged contextual inquiry process. Data-driven vs. model-driven inquiry. Organizing concepts: work roles and flow model. Creating and managing work activity notes. Constructing your work activity affinity diagram (WAAD). Abridged contextual analysis process. History of affinity diagrams.

UNIT III DESIGN THINKING, IDEATION, AND SKETCHING

9

Design-informing models: second span of the bridge . Some general "how to" suggestions. A New example domain: slideshow presentations. User models. Usage models. Work environment models. Barrier summaries. Model consolidation. Protecting your sources. Abridged methods for design-informing models extraction. Design paradigms. Design thinking. Design perspectives. User personas. Ideation. Sketching

UNIT IV UX GOALS, METRICS, AND TARGETS

8

Introduction. UX goals. UX target tables. Work roles, user classes, and UX goals. UX measures. Measuring instruments. UX metrics. Baseline level. Target level. Setting levels. Observed results. Practical tips and cautions for creating UX targets. How UX targets help manage the user experience engineering process.

UNIT V ANALYSING USER EXPERIENCE

10

Sharpening Your Thinking Tools. UX Research and Strength of Evidence. Agile Personas. How to Prioritize Usability Problems. Creating Insights, Hypotheses and Testable Design Ideas. How to Manage Design Projects with User Experience Metrics. Two Measures that Will Justify Any Design Change. Evangelizing UX Research. How to Create a User Journey Map. Generating Solutions to Usability Problems. Building UX Research Into the Design Studio Methodology. Dealing with Common objections to UX Research. The User Experience Debrief Meeting. Creating a User Experience Dashboard.

SUGGESTED ACTIVITIES:

- 1: Hands on Design Thinking process for a product
- 2: Defining the Look and Feel of any new Project
- 3: Create a Sample Pattern Library for that product (Mood board, Fonts, Colors based on UI principles)
- 4: Identify a customer problem to solve.
- 5: Conduct end-to-end user research - User research, creating personas, Ideation process (User stories, Scenarios), Flow diagrams, Flow Mapping

TOTAL : 45 PERIODS

COURSE OUTCOMES:

- CO1:** Build UI for user Applications
- CO2:** Use the UI Interaction behaviors and principles
- CO3:** Evaluate UX design of any product or application
- CO4:** Demonstrate UX Skills in product development
- CO5:** Implement Sketching principles

REFERENCES

1. UX for Developers: How to Integrate User-Centered Design Principles Into Your Day-to-Day Development Work, Westley Knight. Apress, 2018
2. The UX Book: Process and Guidelines for Ensuring a Quality User Experience, Rex Hartson, Pardha Pyla. Morgan Kaufmann, 2012
3. UX Fundamentals for Non-UX Professionals: User Experience Principles for Managers, Writers, Designers, and Developers, Edward Stull. Apress, 2018
4. Lean UX: Designing Great Products with Agile Teams, Gothelf, Jeff, Seiden, and Josh. O'Reilly Media, 2016
5. Designing UX: Prototyping: Because Modern Design is Never Static, Ben Coleman, and Dan Goodwin. SitePoint, 2017

COURSE OBJECTIVES:

- To understand the basics of GPU architectures
- To understand CPU GPU Program Partitioning
- To write programs for massively parallel processors
- To understand the issues in mapping algorithms for GPUs
- To introduce different GPU programming models

UNIT I GPU ARCHITECTURE 9

Evolution of GPU architectures - Understanding Parallelism with GPU –Typical GPU Architecture - CUDA Hardware Overview - Threads, Blocks, Grids, Warps, Scheduling - Memory Handling with CUDA: Shared Memory, Global Memory, Constant Memory and Texture Memory.

UNIT II CUDA PROGRAMMING 9

Using CUDA - Multi GPU - Multi GPU Solutions - Optimizing CUDA Applications: Problem Decomposition, Memory Considerations, Transfers, Thread Usage, Resource Contentions.

UNIT III PROGRAMMING ISSUES 9

Common Problems: CUDA Error Handling, Parallel Programming Issues, Synchronization, Algorithmic Issues, Finding and Avoiding Errors.

UNIT IV OPENCL BASICS 9

OpenCL Standard – Kernels – Host Device Interaction – Execution Environment – Memory Model – Basic OpenCL Examples.

UNIT V ALGORITHMS ON GPU 9

Parallel Patterns: Convolution, Prefix Sum, Sparse Matrix - Matrix Multiplication - Programming Heterogeneous Cluster.

SUGGESTED ACTIVITIES:

1. Debugging Lab
2. Performance Lab
3. Launching Nsight
4. Running Performance Analysis
5. Understanding Metrics
6. NVIDIA Visual Profiler
7. Matrix Transpose Optimization
8. Reduction Optimization

TOTAL : 45 PERIODS**COURSE OUTCOMES:****CO1:** Describe GPU Architecture**CO2:** Write programs using CUDA, identify issues and debug them**CO3:** Implement efficient algorithms in GPUs for common application kernels, such as matrix multiplication**CO4:** Write simple programs using OpenCL**CO5:** Identify efficient parallel programming patterns to solve problems

REFERENCES

1. Shane Cook, CUDA Programming: "A Developer's Guide to Parallel Computing with GPUs (Applications of GPU Computing), First Edition, Morgan Kaufmann, 2012.
2. David R. Kaeli, Perhaad Mistry, Dana Schaa, Dong Ping Zhang, "Heterogeneous computing with OpenCL, 3rd Edition, Morgan Kauffman, 2015.
3. Nicholas Wilt, "CUDA Handbook: A Comprehensive Guide to GPU Programming, Addison - Wesley, 2013.
4. Jason Sanders, Edward Kandrot, "CUDA by Example: An Introduction to General Purpose GPU Programming, Addison - Wesley, 2010.
5. David B. Kirk, Wen-mei W. Hwu, Programming Massively Parallel Processors - A Hands-on Approach, Third Edition, Morgan Kaufmann, 2016.
6. http://www.nvidia.com/object/cuda_home_new.html
7. <http://www.openCL.org>

MP4094

WEB SERVICES AND API DESIGN

L T P C

3 0 0 3

COURSE OBJECTIVES:

- To learn the basics of Web service.
- To become familiar with the Web Services building blocks
- To learn to work with RESTful web services.
- To implement the RESTful web services.
- To understand resource oriented Architecture.

UNIT I INTRODUCTION TO WEB SERVICE

9

Overview – Web service-Architecture – Service-Oriented Architecture (SOA), Architecting Web Services: Web Services Technology Stack, Logical Architectural View, Deployment Architectural View, and Process Architectural View.

UNIT II WEB SERVICE BUILDING BLOCKS

9

Introduction to SOAP: SOAP Syntax- Sending SOAP Messages - SOAP Implementations - Introduction to WSDL: WSDL Syntax - SOAP Binding - WSDL Implementations - Introduction to UDDI: The UDDI API - Implementations - The Future of UDDI

UNIT III RESTFUL WEB SERVICES

9

Programmable Web - HTTP: Documents in Envelopes - Method Information - Scoping Information - The Competing Architectures - Technologies on the Programmable Web -Leftover Terminology - Writing Web Service Clients: The Sample Application - Making the Request: HTTP Libraries - Processing the Response: XML Parsers - JSON Parsers: Handling Serialized Data - Clients Made Easy with WADL.

UNIT IV IMPLEMENTATION OF RESTFUL WEB SERVICES

9

Introducing the Simple Storage Service - Object-Oriented Design of S3 - Resources - HTTP Response Codes Resource- URIs - Addressability - Statelessness - Representations - Links and Connectedness - The Uniform Interface – Spring Web Services – Spring MVC Components - Spring Web Flow - A Service Implementation using Spring Data REST.

UNIT V RESOURCE ORIENTED ARCHITECTURE

9

Resource- URIs - Addressability - Statelessness - Representations - Links and Connectedness - The Uniform Interface- Designing Read-Only Resource-Oriented Services : Resource Design - Turning Requirements Into Read-Only Resources - Figure Out the Data Set- Split the Data Set into Resources- Name the Resources - Design Representation- Link the Resources to Each Other- The HTTP Response

TOTAL : 45 PERIODS

COURSE OUTCOMES:

CO1: Explain how to write XML documents.

CO2: Apply the web service building blocks such as SOAP, WSDL and UDDI

CO3: Describe the RESTful web services.

CO4: Implement the RESTful web service web service with Spring Boot MVC

CO5: Discuss Resource-oriented Architecture.

REFERENCES

1. Leonard Richardson and Sam Ruby, RESTful Web Services, O'Reilly Media, 2007
2. McGovern, et al., "Java Web Services Architecture", Morgan Kaufmann Publishers,2005.
3. Lindsay Bassett, Introduction to JavaScript Object Notation, O'Reilly Media, 2015
4. Craig Walls, "Spring in Action, Fifth Edition", Manning Publications, 2018
5. Raja CSP Raman, Ludovic Dewailly, "Building A RESTful Web Service with Spring 5", Packt Publishing, 2018 .
6. Bogunuva Mohanram Balachandar, "Restful Java Web Services, Third Edition: A pragmatic guide to designing and building RESTful APIs using Java", Ingram short title, 3rd Edition, 2017.
7. Mario-Leander Reimer, "Building RESTful Web Services with Java EE 8: Create modern RESTful web services with the Java EE 8 API", Packt publishing, 2018.

IF4073

DEVOPS AND MICROSERVICES

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

COURSE OBJECTIVES:

- To learn the basic concepts and terminology of DevOps
- To gain knowledge on Devops platform
- To understand building and deployment of code
- To be familiar with DevOps automation tools
- To learn basics of MLOps

UNIT I INTRODUCTION

9+6

Software Engineering - traditional and Agile process models - DevOps -Definition - Practices - DevOps life cycle process - need for DevOps –Barriers

UNIT II DEVOPS PLATFORM AND SERVICES

9+6

Cloud as a platform - IaaS, PaaS, SaaS - Virtualization - Containers –Supporting Multiple Data Centers - Operation Services - Hardware provisioning- software Provisioning - IT services - SLA - capacity planning - security - Service Transition - Service Operation Concepts.

UNIT III BUILDING , TESTING AND DEPLOYMENT

9+6

Microservices architecture - coordination model - building and testing - Deployment pipeline - Development and Pre-commit Testing -Build and Integration Testing - continuous integration - monitoring - security - Resources to Be Protected - Identity Management

UNIT IV DEVOPS AUTOMATION TOOLS

9+6

Infrastructure Automation- Configuration Management - Deployment Automation - Performance Management - Log Management -Monitoring.

UNIT V MLOPS

9+6

MLOps - Definition - Challenges -Developing Models - Deploying to production - Model Governance - Real world examples

SUGGESTED ACTIVITIES:

- 1: Creating a new Git repository, cloning existing repository, Checking changes into a Git repository, Pushing changes to a Git remote, Creating a Git branch
- 2: Installing Docker container on windows/Linux, issuing docker commands
- 3: Building Docker Images for Python Application
- 4: Setting up Docker and Maven in Jenkins and First Pipeline Run
- 5: Running Unit Tests and Integration Tests in Jenkins Pipelines

TOTAL:75 PERIODS

COURSE OUTCOMES:

CO1: Implement modern software Engineering process

CO2: work with DevOps platform

CO3: build, test and deploy code

CO4: Explore DevOps tools

CO5: Correlate MLOps concepts with real time examples

REFERENCES

1. Len Bass, Ingo Weber and Liming Zhu, “DevOps: A Software Architect’s Perspective”, Pearson Education, 2016
2. Joakim Verona - “Practical DevOps” - Packet Publishing , 2016
3. Viktor Farcic -”The DevOps 2.1 Toolkit: Docker Swarm” - Packet Publishing, 2017
4. Mark Treveil, and the Dataiku Team-”Introducing MLOps” - O’Reilly Media- 2020

IF4071

DEEP LEARNING

L T P C

3 0 2 4

COURSE OBJECTIVES:

- Develop and Train Deep Neural Networks.
- Develop a CNN, R-CNN, Fast R-CNN, Faster-R-CNN, Mask-RCNN for detection and recognition
- Build and train RNNs, work with NLP and Word Embeddings
- The internal structure of LSTM and GRU and the differences between them
- The Auto Encoders for Image Processing

UNIT I DEEP LEARNING CONCEPTS 6

Fundamentals about Deep Learning. Perception Learning Algorithms. Probabilistic modelling. Early Neural Networks. How Deep Learning different from Machine Learning. Scalars. Vectors. Matrixes, Higher Dimensional Tensors. Manipulating Tensors. Vector Data. Time Series Data. Image Data. Video Data.

UNIT II NEURAL NETWORKS 9

About Neural Network. Building Blocks of Neural Network. Optimizers. Activation Functions. Loss Functions. Data Pre-processing for neural networks, Feature Engineering. Overfitting and Underfitting. Hyperparameters.

UNIT III CONVOLUTIONAL NEURAL NETWORK 10

About CNN. Linear Time Invariant. Image Processing Filtering. Building a convolutional neural network. Input Layers, Convolution Layers. Pooling Layers. Dense Layers. Backpropagation Through the Convolutional Layer. Filters and Feature Maps. Backpropagation Through the Pooling Layers. Dropout Layers and Regularization. Batch Normalization. Various Activation Functions. Various Optimizers. LeNet, AlexNet, VGG16, ResNet. Transfer Learning with Image Data. Transfer Learning using Inception Oxford VGG Model, Google Inception Model, Microsoft ResNet Model. R-CNN, Fast R-CNN, Faster R-CNN, Mask-RCNN, YOLO

UNIT VI NATURAL LANGUAGE PROCESSING USING RNN 10

About NLP & its Toolkits. Language Modeling . Vector Space Model (VSM). Continuous Bag of Words (CBOW). Skip-Gram Model for Word Embedding. Part of Speech (PoS) Global Co-occurrence Statistics-based Word Vectors. Transfer Learning. Word2Vec. Global Vectors for Word Representation GloVe. Backpropagation Through Time. Bidirectional RNNs (BRNN) . Long Short Term Memory (LSTM). Bi-directional LSTM. Sequence-to-Sequence Models (Seq2Seq). Gated recurrent unit GRU.

UNIT V DEEP REINFORCEMENT & UNSUPERVISED LEARNING 10

About Deep Reinforcement Learning. Q-Learning. Deep Q-Network (DQN). Policy Gradient Methods. Actor-Critic Algorithm. About Autoencoding. Convolutional Auto Encoding. Variational Auto Encoding. Generative Adversarial Networks. Autoencoders for Feature Extraction. Auto Encoders for Classification. Denoising Autoencoders. Sparse Autoencoders

LIST OF EXPERIMENTS 30

- 1: Feature Selection from Video and Image Data
- 2: Image and video recognition
- 3: Image Colorization
- 4: Aspect Oriented Topic Detection & Sentiment Analysis
- 5: Object Detection using Autoencoder

COURSE OUTCOMES:

CO1: Feature Extraction from Image and Video Data

CO2: Implement Image Segmentation and Instance Segmentation in Images

CO3: Implement image recognition and image classification using a pretrained network (Transfer Learning)

CO4: Traffic Information analysis using Twitter Data

CO5: Autoencoder for Classification & Feature Extraction

TOTAL: 45+30=75 PERIODS

REFERENCES

1. Deep Learning A Practitioner's Approach Josh Patterson and Adam Gibson O'Reilly Media, Inc.2017
2. Learn Keras for Deep Neural Networks, Jojo Moolayil, Apress,2018
3. Deep Learning Projects Using TensorFlow 2, Vinita Silaparasetty, Apress, 2020
4. Deep Learning with Python, FRANÇOIS CHOLLET, MANNING SHELTER ISLAND,2017
5. Pro Deep Learning with TensorFlow, Santanu Pattanayak, Apress,2017

CP4072

BLOCKCHAIN TECHNOLOGIES

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

COURSE OBJECTIVES:

- This course is intended to study the basics of Blockchain technology.
- During this course the learner will explore various aspects of Blockchain technology like application in various domains.
- By implementing, learners will have idea about private and public Blockchain, and smart contract.

UNIT I INTRODUCTION OF CRYPTOGRAPHY AND BLOCKCHAIN 9

Introduction to Blockchain, Blockchain Technology Mechanisms & Networks, Blockchain Origins, Objective of Blockchain, Blockchain Challenges, Transactions and Blocks, P2P Systems, Keys as Identity, Digital Signatures, Hashing, and public key cryptosystems, private vs. public Blockchain.

UNIT II BITCOIN AND CRYPTOCURRENCY 9

Introduction to Bitcoin, The Bitcoin Network, The Bitcoin Mining Process, Mining Developments, Bitcoin Wallets, Decentralization and Hard Forks, Ethereum Virtual Machine (EVM), Merkle Tree, Double-Spend Problem, Blockchain and Digital Currency, Transactional Blocks, Impact of Blockchain Technology on Cryptocurrency.

UNIT III INTRODUCTION TO ETHEREUM 9

Introduction to Ethereum, Consensus Mechanisms, Metamask Setup, Ethereum Accounts, Transactions, Receiving Ethers, Smart Contracts.

UNIT-IV INTRODUCTION TO HYPERLEDGER AND SOLIDITY PROGRAMMING 10

Introduction to Hyperledger, Distributed Ledger Technology & its Challenges, Hyperledger & Distributed Ledger Technology, Hyperledger Fabric, Hyperledger Composer. Solidity - Language of Smart Contracts, Installing Solidity & Ethereum Wallet, Basics of Solidity, Layout of a Solidity Source File & Structure of Smart Contracts, General Value Types.

UNIT V BLOCKCHAIN APPLICATIONS 8

Internet of Things, Medical Record Management System, Domain Name Service and Future of Blockchain, Alt Coins.

TOTAL: 45 PERIODS

LIST OF EXPERIMENTS:

1. Create a Simple Blockchain in any suitable programming language.
2. Use Geth to Implement Private Ethereum Block Chain.
3. Build Hyperledger Fabric Client Application.
4. Build Hyperledger Fabric with Smart Contract.
5. Create Case study of Block Chain being used in illegal activities in real world.
6. Using Python Libraries to develop Block Chain Application.

TOTAL: 30 PERIODS

SUPPLEMENTARY RESOURCES:

- NPTEL online course : <https://nptel.ac.in/courses/106/104/106104220/#>
- Udemy: <https://www.udemy.com/course/build-your-blockchain-az/>
- EDUXLABS Online training :<https://eduxlabs.com/courses/blockchain-technology-training/?tab=tab-curriculum>

TOTAL: 75 PERIODS

COURSE OUTCOMES:

After the completion of this course, student will be able to

CO1: Understand and explore the working of Blockchain technology

CO2: Analyze the working of Smart Contracts

CO3: Understand and analyze the working of Hyperledger

CO4: Apply the learning of solidity to build de-centralized apps on Ethereum

CO5: Develop applications on Blockchain

REFERENCES:

1. Imran Bashir, "Mastering Blockchain: Distributed Ledger Technology, Decentralization, and Smart Contracts Explained", Second Edition, Packt Publishing, 2018.
2. Narayanan, J. Bonneau, E. Felten, A. Miller, S. Goldfeder, "Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction" Princeton University Press, 2016
3. Antonopoulos, Mastering Bitcoin, O'Reilly Publishing, 2014. .
4. Antonopoulos and G. Wood, "Mastering Ethereum: Building Smart Contracts and Dapps", O'Reilly Publishing, 2018.
5. D. Drescher, Blockchain Basics. Apress, 2017.

IF 4291

FULL STACK WEB APPLICATION DEVELOPMENT

L T P C

3 0 2 4

COURSE OBJECTIVES:

- Develop TypeScript Application
- Develop Single Page Application (SPA)
- Able to communicate with a server over the HTTP protocol
- Learning all the tools need to start building applications with Node.js
- Implement the Full Stack Development using MEAN Stack

UNIT I FUNDAMENTALS & TYPESCRIPT LANGUAGE 10

Server-Side Web Applications. Client-Side Web Applications. Single Page Application. About TypeScript. Creating TypeScript Projects. TypeScript Data Types. Variables. Expression and Operators. Functions. OOP in Typescript. Interfaces. Generics. Modules. Enums. Decorators. Enums. Iterators. Generators.

UNIT II ANGULAR 10

About Angular. Angular CLI. Creating an Angular Project. Components. Components Interaction. Dynamic Components. Angular Elements. Angular Forms. Template Driven Forms. Property, Style, Class and Event Binding. Two way Bindings. Reactive Forms. Form Group. Form Controls. About Angular Router. Router Configuration. Router State. Navigation Pages. Router Link. Query Parameters. URL matching. Matching Strategies. Services. Dependency Injection. HttpClient. Read Data from the Server. CRUD Operations. Http Header Operations. Intercepting requests and responses.

UNIT III NODE.Js 10

About Node.js. Configuring Node.js environment. Node Package Manager NPM. Modules. Asynchronous Programming. Call Stack and Event Loop. Callback functions. Callback errors. Abstracting callbacks. Chaining callbacks. File System. Synchronous vs. asynchronous I/O. Path and directory operations. File Handle. File Synchronous API. File Asynchronous API. File Callback API. Timers. Scheduling Timers. Timers Promises API. Node.js Events. Event Emitter. Event Target and Event API. Buffers. Buffers and TypedArrays. Buffers and iteration. Using buffers for binary data. Flowing vs. non-flowing streams. JSON.

UNIT IV EXPRESS.Js 7

Express.js. How Express.js Works. Configuring Express.js App Settings. Defining Routes. Starting the App. Express.js Application Structure. Configuration, Settings. Middleware. body-parser. cookie-parser. express-session. response-time. Template Engine. Jade. EJS. Parameters. Routing. router.route(path). Router Class. Request Object. Response Object. Error Handling. RESTful.

UNIT V MONGODB 8

Introduction to MongoDB. Documents. Collections. Subcollections. Database. Data Types. Dates. Arrays. Embedded Documents. CRUD Operations. Batch Insert. Insert Validation. Querying The Documents. Cursors. Indexing. Unique Indexes. Sparse Indexes. Special Index and Collection Types. Full-Text Indexes. Geospatial Indexing. Aggregation framework.

LIST OF EXPERIMENTS 30

- 1: Accessing the Weather API from Angular
- 2: Accessing the Stock Market API from Angular
- 3: Call the Web Services of Express.js From Angular
- 4: Read the data in Node.js from MongoDB
- 5: CRUD operation in MongoDB using Angular

COURSE OUTCOMES:

- CO1:** Develop basic programming skills using Javascript
- CO2:** Implement a front-end web application using Angular.
- CO3:** Will be able to create modules to organise the server

CO4: Build RESTful APIs with Node, Express and MongoDB with confidence.

CO5: Will learn to Store complex, relational data in MongoDB using Mongoose

TOTAL: 45 + 30=75 PERIODS

REFERENCES

1. Adam Freeman, Essential TypeScript, Apress, 2019
2. Mark Clow, Angular Projects, Apress, 2018
3. Alex R. Young, Marc Harter, Node.js in Practice, Manning Publication, 2014
4. Pro Express.js, Azat Mardan, Apress, 2015
5. MongoDB in Action, Kyle Banker, Peter Bakkum, Shaun Verch, Douglas Garrett, Tim Hawkins, Manning Publication, Second edition, 2016

SE4073

EMBEDDED SOFTWARE DEVELOPMENT

L T P C
3 0 2 4

COURSE OBJECTIVES:

- To understand the architecture of embedded processor, microcontroller, and peripheral devices.
- To interface memory and peripherals with embedded systems.
- To study the embedded network environment.
- To understand challenges in Real time operating systems.
- To study, analyze and design applications on embedded systems.

UNIT I EMBEDDED PROCESSORS

9+6

Embedded Computers - Characteristics of Embedded Computing Applications - Challenges in Embedded Computing System Design - Embedded System Design Process- Formalism for System Design - Structural Description - Behavioural Description - ARM Processor - Intel ATOM Processor.

UNIT II EMBEDDED COMPUTING PLATFORM

9+6

CPU Bus Configuration - Memory Devices and Interfacing - Input/Output Devices and Interfacing - System Design - Development and Debugging – Emulator – Simulator - JTAG Design Example – Alarm Clock - Analysis and Optimization of Performance - Power and Program Size.

UNIT III EMBEDDED NETWORK ENVIRONMENT

9+6

Distributed Embedded Architecture - Hardware And Software Architectures - Networks for Embedded Systems - I2C - CAN Bus - SHARC Link Supports – Ethernet – Myrinet – Internet - Network-based Design - Communication Analysis - System Performance Analysis - Hardware Platform Design - Allocation and Scheduling - Design Example - Elevator Controller.

UNIT IV REAL-TIME CHARACTERISTICS

9+6

Clock Driven Approach - Weighted Round Robin Approach - Priority Driven Approach - Dynamic versus Static Systems - Effective Release Times and Deadlines - Optimality of the Earliest Deadline First (EDF) Algorithm - Challenges in Validating Timing Constraints in Priority Driven Systems - Off-Line versus On-Line Scheduling.

UNIT V SYSTEM DESIGN TECHNIQUES

9+6

Design Methodologies - Requirement Analysis – Specification - System Analysis and Architecture Design - Quality Assurance - Design Examples - Telephone PBX - Ink jet printer - Personal Digital Assistants - Set-Top Boxes.

SUGGESTED ACTIVITIES:

1. Study of ARM evaluation system
2. Interfacing ADC and DAC.
3. Interfacing LED and PWM.
4. Interfacing real time clock and serial port.
5. Interfacing keyboard and LCD.
6. Interfacing EPROM and interrupt.
7. Principles of Mailbox.
8. Interrupt performance characteristics of ARM and FPGA.
9. Flashing of LEDS.
10. Interfacing stepper motor and temperature sensor.

COURSE OUTCOMES:

CO1: Understand different architectures of embedded processor, microcontroller and peripheral devices. Interface memory and peripherals with embedded systems.

CO2: Interface memory and peripherals with embedded systems.

CO3: Work with embedded network environment.

CO4: Understand challenges in Real time operating systems.

CO5: Design and analyze applications on embedded systems.

TOTAL:45+30=75 PERIODS

REFERENCES

1. Adrian McEwen, Hakim Cassimally, "Designing the Internet of Things" Wiley Publication, First edition, 2013
2. Andrew N Sloss, D. Symes, C. Wright, Arm system developers guide, Morgan Kauffman/Elsevier, 2006.
3. ArshdeepBahga, Vijay Madisetti, " Internet of Things: A Hands-on-Approach" VPT First Edition, 2014
4. C. M. Krishna and K. G. Shin, "Real-Time Systems , McGraw-Hill, 1997
5. Frank Vahid and Tony Givargis, "Embedded System Design: A Unified Hardware/Software Introduction, John Wiley & Sons.1999
6. Jane.W.S. Liu, "Real-Time systems, Pearson Education Asia,2000
7. Michael J. Pont, "Embedded C, Pearson Education, 2007.
8. Muhammad Ali Mazidi , SarmadNaimi , SepehrNaimi, "The AVR Microcontroller and Embedded Systems: Using Assembly and C" Pearson Education, First edition, 2014
9. Steve Heath, "Embedded System Design, Elsevier, 2005
10. Wayne Wolf, "Computers as Components:Principles of Embedded Computer System Design, Elsevier, 2006.

COURSE OBJECTIVES:

- Learn Depth Concept of GO Programming
- How to develop Smart Contracts
- How to Deploy Smart Contracts
- Front end Development using Angular
- Implementing Bitcoin Network

UNIT I	GETTING STARTED WITH GO PROGRAMMING	8
Centralized vs Decentralized Systems Centralized Systems Decentralized Systems. Decentralized Data. Decentralized Wealth. Decentralized Identity. Decentralized Computing. Decentralized Bandwidth. Decentralized Markets for Decentralized Assets. About Go Language. The Terminal. Environment. Go. Your First Program. Variables & Data Types. Control Structures. Collection Frameworks. Functions. Structs and Interfaces. Packages. Hashes and Cryptography Packages. Servers Packages. Concurrency. Goroutines. Channels. Channel Direction. Select. Buffered Channels. The sync package. Synchronizing with mutex locks. Synchronizing access to composite values. Concurrency barriers with sync. WaitGroup. Data IO.		
UNIT II	BUILDING DISTRIBUTED APPLICATIONS IN GIN	8
Installing and configuring Gin. Dependency management in Golang. Writing a custom HTTP handler. Exploring API functionality. Defining the data model. HTTP endpoints. Implementing HTTP routes & Methods. Managing Data Persistence with MongoDB. Authentication & Authorization. Developing and Deploying Web Application using Gin		
UNIT III	SMART CONTRACTS USING SOLIDITY & GO	10
The CAP theorem. Consensus in distributed systems. Understanding the hash function and the Merkle tree. Operations using Solidity. Control Structures. Smart contract on a private blockchain. Design of DAO. Class properties of a contract. Expression and control structures. State variables. Functions & its Modifiers. Events. Implementing funding limit with inheritance. Making a contract abstract.		
UNIT IV	DEVELOPING DAPPS	9
What Is a DApp?. DApp architecture. Backend (Smart Contract). Frontend (Web User Interface). Data Storage. Inter-Planetary File System (IPFS). Swarm. Developing a Cryptocurrency. Building Your Dapp. Routing. Data Storage and Retrieval. Exploring the Truffle suite. Learning Solidity's advanced features. Contract testing and debugging. Ethereum DApp with Angular.		
UNIT V	BITCOIN NETWORK	10
The Bitcoin Network. Network Discovery for a New Node. Bitcoin Transactions. Consensus and Block Mining. Block Propagation. Bitcoin payments. Bitcoin client. Bitcoin programming. Running a Blockchain Node. Create a Bitcoin Miner. Create a NEO Bookkeeping Node. Create an EOS Block Producer. Bitcoin Core API. Serialized Blocks. Block Header. Block Version. Bitcoin Wallet.		

LIST OF EXPERIMENTS

30

- 1: Developing Purchase Order DApp
- 2: Designing a Voting DApp
- 3: Designing and Deploying Vaccine Production using DApp
- 4: Developing Auction DApp
- 5: Developing Property Registration DApp

COURSE OUTCOMES:

- CO1:** Learn How to Compile and Deploy Solidity
CO2: Use Golang to Connect to Ethereum
CO3: Deploy Ethereum Smart Contracts Using Golang
CO4: Develop DApp using Angular
CO5: Develop Bitcoin Application

TOTAL : 45+30=75 PERIODS

REFERENCES

1. Caleb Doxsey, "Introducing Go", O'Reilly Media, 2016
2. Vladimir Vivien, "Learning Go Programming", Packt Publishing, 2016
3. Siraj Raval, "Decentralized Applications", O'Reilly Media, 2016
4. Mohamed Labouardy, "Building Distributed Applications in Gin", Packt Publishing, 2021
5. Chris Dannen, "Introducing Ethereum and Solidity", Apress, 2017

CP4291

INTERNET OF THINGS

L T P C
3 0 2 4

COURSE OBJECTIVES:

- To Understand the Architectural Overview of IoT
- To Understand the IoT Reference Architecture and Real World Design Constraints
- To Understand the various IoT levels
- To understand the basics of cloud architecture
- To gain experience in Raspberry PI and experiment simple IoT application on it

UNIT I INTRODUCTION

9+6

Internet of Things- Domain Specific IoTs - IoT and M2M-Sensors for IoT Applications–Structure of IoT– IoT Map Device- IoT System Management with NETCONF-YANG

UNIT II IoT ARCHITECTURE, GENERATIONS AND PROTOCOLS

9+6

IETF architecture for IoT - IoT reference architecture -First Generation – Description & Characteristics–Advanced Generation – Description & Characteristics–Integrated IoT Sensors – Description & Characteristics

UNIT III IoT PROTOCOLS AND TECHNOLOGY

9+6

SCADA and RFID Protocols - BACNet Protocol -Zigbee Architecture - 6LowPAN - CoAP -Wireless Sensor Structure–Energy Storage Module–Power Management Module–RF Module–Sensing Module

UNIT IV CLOUD ARCHITECTURE BASICS

9+6

The Cloud types; IaaS, PaaS, SaaS.- Development environments for service development; Amazon, Azure, Google Appcloud platform in industry

UNIT V IOT PROJECTS ON RASPBERRY PI

9+6

Building IOT with RASPBERRY PI- Creating the sensor project - Preparing Raspberry Pi - Clayster libraries – Hardware Interacting with the hardware - Interfacing the hardware- Internal representation of sensor values - Persisting data - External representation of sensor values - Exporting sensor data

SUGGESTED ACTIVITIES:

1. Develop an application for LED Blink and Pattern using arduino or Raspberry Pi
2. Develop an application for LED Pattern with Push Button Control using arduino or Raspberry Pi
3. Develop an application for LM35 Temperature Sensor to display temperature values using arduino or Raspberry Pi
4. Develop an application for Forest fire detection end node using Raspberry Pi device and sensor
5. Develop an application for home intrusion detection web application
6. Develop an application for Smart parking application using python and Django for web application

COURSE OUTCOMES:

CO1: Understand the various concept of the IoT and their technologies

CO2: Develop the IoT application using different hardware platforms

CO3: Implement the various IoT Protocols

CO4: Understand the basic principles of cloud computing

CO5: Develop and deploy the IoT application into cloud environment

TOTAL: 75 PERIODS

REFERENCES:

1. Arshdeep Bahga, Vijay Madiseti, Internet of Things: A hands-on approach, Universities Press, 2015
2. Dieter Uckelmann, Mark Harrison, Florian Michahelles (Eds), Architecting the Internet of Things, Springer, 2011
3. Peter Waher, 'Learning Internet of Things', Packt Publishing, 2015
4. Ovidiu Vermesan Peter Friess, 'Internet of Things – From Research and Innovation to Market Deployment', River Publishers, 2014
5. N. Ida, Sensors, Actuators and Their Interfaces: A Multidisciplinary Introduction, 2nd Edition Scitech Publishers, 202014
6. Reese, G. (2009). Cloud Application Architectures: Building Applications and Infrastructure in the Cloud. Sebastopol, CA: O'Reilly Media, Inc. (2009)

COURSE OBJECTIVES:

- To study about Fundamental Concept and Components of Virtual Reality
- To study about Interactive Techniques in Virtual Reality
- To study about Visual Computation in Virtual Reality
- To study about Augmented and Mixed Reality and Its Applications
- To know about I/O Interfaces and its functions.

UNIT I INTRODUCTION TO VIRTUAL REALITY**9**

Introduction, Fundamental Concept and Components of Virtual Reality. Primary Features and Present Development on Virtual Reality. Computer graphics, Real time computer graphics, Flight Simulation, Virtual environment requirement, benefits of virtual reality, Historical development of VR, Scientific Landmark 3D Computer Graphics: Introduction, The Virtual world space, positioning the virtual observer, the perspective projection, human vision, stereo perspective projection, 3D clipping, Colour theory, Simple 3D modelling, Illumination models, Reflection models, Shading algorithms, Radiosity, Hidden Surface Removal, Realism Stereographic image.

Suggested Activities:

- Flipped classroom on uses of MR applications.
- Videos – Experience the virtual reality effect.
- Assignment on comparison of VR with traditional multimedia applications.

Suggested Evaluation Methods:

- Tutorial – Applications of MR.
- Quizzes on the displayed video and the special effects

UNIT II INTERACTIVE TECHNIQUES IN VIRTUAL REALITY**9**

Introduction, from 2D to 3D, 3D spaces curves, 3D boundary representation Geometrical Transformations: Introduction, Frames of reference, Modeling transformations, Instances, Picking, Flying, Scaling the VE, Collision detection Generic VR system: Introduction, Virtual environment, Computer environment, VR technology, Model of interaction, VR Systems.

Suggested Activities:

- Flipped classroom on modeling three dimensional objects.
- External learning – Collision detection algorithms.
- Practical – Creating three dimensional models.

Suggested Evaluation Methods:

- Tutorial – Three dimensional modeling techniques.
- Brainstorming session on collision detection algorithms.
- Demonstration of three dimensional scene creation.

UNIT III VISUAL COMPUTATION IN VIRTUAL REALITY**9**

Animating the Virtual Environment: Introduction, The dynamics of numbers, Linear and Nonlinear interpolation, the animation of objects, linear and non-linear translation, shape & object inbetweening, free from deformation, particle system. Physical Simulation: Introduction, Objects falling in a gravitational field, Rotating wheels, Elastic collisions, projectiles, simple pendulum, springs, Flight dynamics of an aircraft.

Suggested Activities:

- External learning – Different types of programming toolkits and Learn different types of available VR applications.
- Practical – Create VR scenes using any toolkit and develop applications.

Suggested Evaluation Methods:

- Tutorial – VR tool comparison.
- Brainstorming session on tools and technologies used in VR.
- Demonstration of the created VR applications.
-

UNIT IV AUGMENTED AND MIXED REALITY

9

Taxonomy, technology and features of augmented reality, difference between AR and VR, Challenges with AR, AR systems and functionality, Augmented reality methods, visualization techniques for augmented reality, wireless displays in educational augmented reality applications, mobile projection interfaces, marker-less tracking for augmented reality, enhancing interactivity in AR environments, evaluating AR systems

Suggested Activities:

- External learning - AR Systems

Suggested Evaluation Methods:

- Brainstorming session different AR systems and environments.

UNIT V I/O INTERFACE IN VR & APPLICATION OF VR

9

Human factors: Introduction, the eye, the ear, the somatic senses. VR Hardware: Introduction, sensor hardware, Head-coupled displays, Acoustic hardware, Integrated VR systems. VR Software: Introduction, Modeling virtual world, Physical simulation, VR toolkits, Introduction to VRML, Input -- Tracker, Sensor, Digitalglobe, Movement Capture, Video-based Input, 3D Menus & 3DScanner etc. Output -- Visual /Auditory / Haptic Devices. VR Technology in Film & TV Production. VR Technology in Physical Exercises and Games. Demonstration of Digital Entertainment by VR.

Suggested Activities:

- External learning – Different types of sensing and tracking devices for creating mixed reality environments.
- Practical – Create MR scenes using any toolkit and develop applications.

Suggested Evaluation Methods:

- Tutorial – Mobile Interface Design.
- Brainstorming session on wearable computing devices and games design.
- Demonstration and evaluation of the developed MR application.

TOTAL: 45 PERIODS

PRACTICALS:

1. Study of tools like Unity, Maya, 3DS MAX, AR toolkit, Vuforia and Blender.
2. Use the primitive objects and apply various projection methods by handling the camera.
3. Download objects from asset stores and apply various lighting and shading effects.
4. Model three dimensional objects using various modeling techniques and apply textures over them.
5. Create three dimensional realistic scenes and develop simple virtual reality enabled mobile applications which have limited interactivity.
6. Add audio and text special effects to the developed application.
7. Develop VR enabled applications using motion trackers and sensors incorporating full haptic interactivity.

8. Develop AR enabled applications with interactivity like E learning environment, Virtual walkthroughs and visualization of historic places.
9. Develop MR enabled simple applications like human anatomy visualization, DNA/RNA structure visualization and surgery simulation.
10. Develop simple MR enabled gaming applications.

TOTAL: 45+30=75 PERIODS

COURSE OUTCOMES:

CO1: Understand the Fundamental Concept and Components of Virtual Reality

CO2: Able to know the Interactive Techniques in Virtual Reality

CO3: Can know about Visual Computation in Virtual Reality

CO4: Able to know the concepts of Augmented and Mixed Reality and Its Applications

CO5: Know about I/O Interfaces and its functions.

REFERENCES

1. Burdea, G. C. and P. Coffet. Virtual Reality Technology, Second Edition. Wiley-IEEE Press, 2003/2006.
2. Alan B. Craig, Understanding Augmented Reality, Concepts and Applications, Morgan Kaufmann, First Edition 2013.
3. Alan Craig, William Sherman and Jeffrey Will, Developing Virtual Reality Applications, Foundations of Effective Design, Morgan Kaufmann, 2009.
4. John Vince, "Virtual Reality Systems ", Pearson Education Asia, 2007.
5. Adams, "Visualizations of Virtual Reality", Tata McGraw Hill, 2000.
6. Grigore C. Burdea, Philippe Coiffet , "Virtual Reality Technology", Wiley Inter Science, 2nd Edition, 2006.
7. William R. Sherman, Alan B. Craig, "Understanding Virtual Reality: Interface, Application and Design", Morgan Kaufmann, 2008

AUDIT COURSES

AX4091

ENGLISH FOR RESEARCH PAPER WRITING

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

COURSE OBJECTIVES:

- Teach how to improve writing skills and level of readability
- Tell about what to write in each section
- Summarize the skills needed when writing a Title
- Infer the skills needed when writing the Conclusion
- Ensure the quality of paper at very first-time submission

UNIT I INTRODUCTION TO RESEARCH PAPER WRITING

6

Planning and Preparation, Word Order, Breaking up long sentences, Structuring Paragraphs and Sentences, Being Concise and Removing Redundancy, Avoiding Ambiguity and Vagueness

UNIT II PRESENTATION SKILLS

6

Clarifying Who Did What, Highlighting Your Findings, Hedging and Criticizing, Paraphrasing and Plagiarism, Sections of a Paper, Abstracts, Introduction

UNIT III TITLE WRITING SKILLS 6
Key skills are needed when writing a Title, key skills are needed when writing an Abstract, key skills are needed when writing an Introduction, skills needed when writing a Review of the Literature, Methods, Results, Discussion, Conclusions, The Final Check

UNIT IV RESULT WRITING SKILLS 6
Skills are needed when writing the Methods, skills needed when writing the Results, skills are needed when writing the Discussion, skills are needed when writing the Conclusions

UNIT V VERIFICATION SKILLS 6
Useful phrases, checking Plagiarism, how to ensure paper is as good as it could possibly be the first-time submission

TOTAL: 30 PERIODS

COURSE OUTCOMES:

- CO1 – Understand that how to improve your writing skills and level of readability
- CO2 – Learn about what to write in each section
- CO3 – Understand the skills needed when writing a Title
- CO4 – Understand the skills needed when writing the Conclusion
- CO5 – Ensure the good quality of paper at very first-time submission

REFERENCES:

1. Adrian Wallwork , English for Writing Research Papers, Springer New York Dordrecht Heidelberg London, 2011
2. Day R How to Write and Publish a Scientific Paper, Cambridge University Press 2006
3. Goldbort R Writing for Science, Yale University Press (available on Google Books) 2006
4. Highman N, Handbook of Writing for the Mathematical Sciences, SIAM. Highman's book 1998.

AX4092

DISASTER MANAGEMENT

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

COURSE OBJECTIVES:

- Summarize basics of disaster
- Explain a critical understanding of key concepts in disaster risk reduction and humanitarian response.
- Illustrate disaster risk reduction and humanitarian response policy and practice from multiple perspectives.
- Describe an understanding of standards of humanitarian response and practical relevance in specific types of disasters and conflict situations.
- Develop the strengths and weaknesses of disaster management approaches

UNIT I INTRODUCTION 6
Disaster: Definition, Factors and Significance; Difference between Hazard And Disaster; Natural and Manmade Disasters: Difference, Nature, Types and Magnitude.

UNIT II REPERCUSSIONS OF DISASTERS AND HAZARDS 6
Economic Damage, Loss of Human and Animal Life, Destruction Of Ecosystem. Natural Disasters: Earthquakes, Volcanisms, Cyclones, Tsunamis, Floods, Droughts And Famines,

Landslides And Avalanches, Man-made disaster: Nuclear Reactor Meltdown, Industrial Accidents, Oil Slicks And Spills, Outbreaks Of Disease And Epidemics, War And Conflicts.

UNIT III DISASTER PRONE AREAS IN INDIA 6

Study of Seismic Zones; Areas Prone To Floods and Droughts, Landslides And Avalanches; Areas Prone To Cyclonic and Coastal Hazards with Special Reference To Tsunami; Post-Disaster Diseases and Epidemics

UNIT IV DISASTER PREPAREDNESS AND MANAGEMENT 6

Preparedness: Monitoring Of Phenomena Triggering a Disaster or Hazard; Evaluation of Risk: Application of Remote Sensing, Data from Meteorological And Other Agencies, Media Reports: Governmental and Community Preparedness.

UNIT V RISK ASSESSMENT 6

Disaster Risk: Concept and Elements, Disaster Risk Reduction, Global and National Disaster Risk Situation. Techniques of Risk Assessment, Global Co-Operation in Risk Assessment and Warning, People's Participation in Risk Assessment. Strategies for Survival

TOTAL : 30 PERIODS

COURSE OUTCOMES:

CO1: Ability to summarize basics of disaster

CO2: Ability to explain a critical understanding of key concepts in disaster risk reduction and humanitarian response.

CO3: Ability to illustrate disaster risk reduction and humanitarian response policy and practice from multiple perspectives.

CO4: Ability to describe an understanding of standards of humanitarian response and practical relevance in specific types of disasters and conflict situations.

CO5: Ability to develop the strengths and weaknesses of disaster management approaches

REFERENCES:

1. Goel S. L., Disaster Administration And Management Text And Case Studies”, Deep& Deep Publication Pvt. Ltd., New Delhi,2009.
2. NishithaRai, Singh AK, “Disaster Management in India: Perspectives, issues and strategies “NewRoyal book Company,2007.
3. Sahni, PardeepEt.Al.,” Disaster Mitigation Experiences And Reflections”, Prentice Hall OfIndia, New Delhi,2001.

AX4093

CONSTITUTION OF INDIA

L T P C

2 0 0 0

COURSE OBJECTIVES:

Students will be able to:

- Understand the premises informing the twin themes of liberty and freedom from a civil rights perspective.
- To address the growth of Indian opinion regarding modern Indian intellectuals' constitutional
- Role and entitlement to civil and economic rights as well as the emergence nation hood in

the early years of Indian nationalism.

- To address the role of socialism in India after the commencement of the Bolshevik Revolution in 1917 and its impact on the initial drafting of the Indian Constitution.

UNIT I HISTORY OF MAKING OF THE INDIAN CONSTITUTION

History, Drafting Committee, (Composition & Working)

UNIT II PHILOSOPHY OF THE INDIAN CONSTITUTION

Preamble, Salient Features

UNIT III CONTOURS OF CONSTITUTIONAL RIGHTS AND DUTIES

Fundamental Rights, Right to Equality, Right to Freedom, Right against Exploitation, Right to Freedom of Religion, Cultural and Educational Rights, Right to Constitutional Remedies, Directive Principles of State Policy, Fundamental Duties.

UNIT IV ORGANS OF GOVERNANCE

Parliament, Composition, Qualifications and Disqualifications, Powers and Functions, Executive, President, Governor, Council of Ministers, Judiciary, Appointment and Transfer of Judges, Qualifications, Powers and Functions.

UNIT V LOCAL ADMINISTRATION

District's Administration head: Role and Importance, □ Municipalities: Introduction, Mayor and role of Elected Representative, CEO, Municipal Corporation. Panchayati raj: Introduction, PRI: Zila Panchayat. Elected officials and their roles, CEO Zila Panchayat: Position and role. Block level: Organizational Hierarchy (Different departments), Village level: Role of Elected and Appointed officials, Importance of grass root democracy.

UNIT VI ELECTION COMMISSION

Election Commission: Role and Functioning. Chief Election Commissioner and Election Commissioners - Institute and Bodies for the welfare of SC/ST/OBC and women.

TOTAL: 30 PERIODS

COURSE OUTCOMES:

Students will be able to:

- Discuss the growth of the demand for civil rights in India for the bulk of Indians before the arrival of Gandhi in Indian politics.
- Discuss the intellectual origins of the framework of argument that informed the conceptualization of social reforms leading to revolution in India.
- Discuss the circumstances surrounding the foundation of the Congress Socialist Party [CSP] under the leadership of Jawaharlal Nehru and the eventual failure of the proposal of direct elections through adult suffrage in the Indian Constitution.
- Discuss the passage of the Hindu Code Bill of 1956.

SUGGESTED READING

1. The Constitution of India, 1950 (Bare Act), Government Publication.
2. Dr. S.N. Busi, Dr. B. R. Ambedkar framing of Indian Constitution, 1st Edition, 2015.
3. M.P. Jain, Indian Constitution Law, 7th Edn., Lexis Nexis, 2014.
4. D.D. Basu, Introduction to the Constitution of India, Lexis Nexis, 2015.

UNIT I	சங்க இலக்கியம்	6
	<ol style="list-style-type: none"> 1. தமிழின் துவக்க நூல் தொல்காப்பியம் - எழுத்து, சொல், பொருள் 2. அகநானூறு (82) - இயற்கை இன்னிசை அரங்கம் 3. குறிஞ்சிப் பாட்டின் மலர்க்காட்சி 4. புறநானூறு (95,195) - போரை நிறுத்திய ஔவையார் 	
UNIT II	அறநெறித் தமிழ்	6
	<ol style="list-style-type: none"> 1. அறநெறி வகுத்த திருவள்ளுவர் - அறம் வலியுறுத்தல், அன்புடைமை, ஒப்புரவறிதல், ஈகை, புகழ் 2. பிற அறநூல்கள் - இலக்கிய மருந்து - ஏலாதி, சிறுபஞ்சமூலம், திரிகடுகம், ஆசாரக்கோவை (தூய்மையை வலியுறுத்தும் நூல்) 	
UNIT III	இரட்டைக் காப்பியங்கள்	6
	<ol style="list-style-type: none"> 1. கண்ணகியின் புரட்சி - சிலப்பதிகார வழக்குரை காதை 2. சமூகசேவை இலக்கியம் மணிமேகலை - சிறைக்கோட்டம் அறக்கோட்டமாகிய காதை 	
UNIT IV	அருள்நெறித் தமிழ்	6
	<ol style="list-style-type: none"> 1. சிறுபாணாற்றுப்படை - பாரி முல்லைக்குத் தேர் கொடுத்தது, பேகன் மயிலுக்குப் போர்வை கொடுத்தது, அதியமான் ஔவைக்கு நெல்லிக்கனி கொடுத்தது, அரசர் பண்புகள் 2. நற்றிணை - அன்னைக்குரிய புன்னை சிறப்பு 3. திருமந்திரம் (617, 618) - இயமம் நியமம் விதிகள் 4. தர்மச்சாலையை நிறுவிய வள்ளலார் 5. புறநானூறு - சிறுவனே வள்ளலானான் 6. அகநானூறு (4) - வண்டு நற்றிணை (11) - நண்டு கலித்தொகை (11) - யானை, புறா 	

ஐந்திணை 50 (27) - மான்
ஆகியவை பற்றிய செய்திகள்

UNIT V

நவீன தமிழ் இலக்கியம்

6

1. உரைநடைத் தமிழ்,
 - தமிழின் முதல் புதினம்,
 - தமிழின் முதல் சிறுகதை,
 - கட்டுரை இலக்கியம்,
 - பயண இலக்கியம்,
 - நாடகம்,
2. நாட்டு விடுதலை போராட்டமும் தமிழ் இலக்கியமும்,
3. சமுதாய விடுதலையும் தமிழ் இலக்கியமும்,
4. பெண் விடுதலையும் விளிம்பு நிலையினரின் மேம்பாட்டில் தமிழ் இலக்கியமும்,
5. அறிவியல் தமிழ்,
6. இணையத்தில் தமிழ்,
7. சுற்றுச்சூழல் மேம்பாட்டில் தமிழ் இலக்கியம்.

TOTAL : 30 PERIODS

தமிழ் இலக்கிய வெளியீடுகள் / புத்தகங்கள்

1. தமிழ் இணைய கல்விக்கழகம் (Tamil Virtual University)
- www.tamilvu.org
2. தமிழ் விக்கிப்பீடியா (Tamil Wikipedia)
- <https://ta.wikipedia.org>
3. தர்மபுர ஆதீன வெளியீடு
4. வாழ்வியல் களஞ்சியம்
- தமிழ்ப் பல்கலைக்கழகம், தஞ்சாவூர்
5. தமிழ்கலைக் களஞ்சியம்
- தமிழ் வளர்ச்சித் துறை (thamilvalarchithurai.com)
6. அறிவியல் களஞ்சியம்
- தமிழ்ப் பல்கலைக்கழகம், தஞ்சாவூர்

PROGRESS THROUGH KNOWLEDGE