SHRI VISHNU ENGINEERING COLLEGE FOR WOMEN :: BHIMAVARAM
(Autonomous)
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Course Structure of M.Tech(Software Engineering)
(With effect from 2022-2023)

I Year - I Semester

S. No.	*Course Code*	*Course Title*	*L*	*T*	*P*	*C*	*IM*	*EM*	*TM*
1	PGCS1T0122	ADVANCED DATA STRUCTURES AND ALGORITHMS	3	-	-	3	40	60	100
2	PGCS1T1122	SOFTWARE REQUIREMENTS AND ESTIMATION	3	-	-	3	40	60	100
3	PGCS1T1222	SOFTWARE PROJECT AND PROCESS MANAGEMENT	3	-	-	3	40	60	100
4	PGCS1T1322	CLOUD COMPUTING	3	-	-	3	40	60	100
5	PGCS1T0522 PGCS1T1422 PGCS1T0722 PGCS1T1522	**ELECTIVE - I** ARTIFICIAL INTELLIGENCE DATA SCIENCE WITH R INTERNET OF THINGS DEVOPS	3	-	-	3	40	60	100
6	PGCS1P0922	ADVANCED DATA STRUCTURES LAB	-	-	3	1.5	40	60	100
7	PGCS1P1622	CLOUD COMPUTING LAB	-	-	3	1.5	40	60	100
		Total	**15**	**0**	**6**	**18**	**280**	**420**	**700**

I Year - II Semester

S. No.	*Course Code*	*Course Title*	*L*	*T*	*P*	*C*	*IM*	*EM*	*TM*
1	PGCS2T0322	FULL STACK DEVELOPMENT	3	-	-	3	40	60	100
2	PGCS2T1422	SOFTWARE ARCHITECTURE AND DESIGN PATTERNS	3	-	-	3	40	60	100
3	PGCS2T1522	SOFTWARE QUALITY ASSURANCE AND TESTING	3	-	-	3	40	60	100
4	PGCS2T0422 PGCS2T1622 PGCS2T1022 PGCS2T1722	**ELECTIVE - II** MACHINE LEARNING NOSQL DATABASES CYBER SECURITY GO PROGRAMMING	3	-	-	3	40	60	100
5	PGCS2T0822 PGCS2T0922 PGCS2T1822 PGCS2T1922	**ELECTIVE - III** DEEP LEARNING BIG DATA TECHNOLOGIES ROBOTIC PROCESS AUTOMATION AUGMENTED REALITY & VIRTUAL REALITY	3	-	-	3	40	60	100
6	PGCS2P1322	FULL STACK DEVELOPMENT LAB	-	-	3	1.5	40	60	100
7	PGCS2P2022	SOFTWARE TESTING AND DESIGN PATTERNS LAB	-	-	3	1.5	40	60	100
		Total	**15**	**0**	**6**	**18**	**280**	**420**	**700**

II Year - I Semester

S. No.	*Course Code*	*Course Title*	*L*	*T*	*P*	*C*	*IM*	*EM*	*TM*
1	PGCS3S0122	COMPREHENSIVE VIVA	-	-	-	2	50	-	50
2		PROJECT WORK	-	-	-	14	-	-	-
		Total				**16**	**50**		**50**

II Year - II Semester

S. No.	*Course Code*	*Course Title*	*L*	*T*	*P*	*C*	*IM*	*EM*	*TM*
1	PGCS4S0122	SEMINAR	-	-	-	2	50	-	50
2	PGCS4J0222	PROJECT WORK	-	-	-	14	-	-	-
		Total				**16**	**50**		**50**

Credits and Total Marks

Year & Semester	*C*	*IM*	*EM*	*TM*
I Year - I Semester	18	280	420	700
I Year - II Semester	18	280	420	700
II Year - I Semester	16	50	-	50
II Year - II Semester	16	50	-	50
Total	**68**	**660**	**840**	**1500**

L – Lectures, T – Tutorials, P – Practicals, C – Credits, IM – Internal Marks, EM – External Marks, TM – Total Marks

I Year

I Semester

ADVANCED DATA STRUCTURES AND ALGORITHMS

Subject Code: PGCS1T0122	**L**	**T**	**P**	**C**
I Year / I Semester	**3**	**0**	**0**	**3**

Prerequisites:
Basic Knowledge on Data Structures, Discrete Mathematics, Advanced Programming Skills and the concept of Abstract Data Types.

Course Objectives:
This course introduces students to a number of highly efficient algorithms and data structures for fundamental computational problems across a variety of areas. Students are also introduced to techniques such as complexity analysis.

Syllabus:

UNIT I: (8 Lectures)
Introduction to Data Structures, Singly Linked Lists, Doubly Linked Lists, Circular Linked Lists-Algorithms.
Stacks and Queues**:** Algorithm Implementation using Linked Lists.

UNIT II: (10 Lectures)
Searching-Linear and Binary Search Methods, Sorting-Bubble Sort, Selection Sort, Insertion Sort, Quick Sort, Merge Sort, Trees- Binary trees, Properties, Representation and Traversals (DFT, BFT), Expression Trees(Infix, prefix, postfix), Graphs-Basic Concepts, Storage Structures and Traversals.

UNIT III: (10 Lectures)
Dictionaries, ADT, The List ADT, Stack ADT, Queue ADT, Hash Table Representation, Hash Functions, Collision Resolution-Separate, Chaining, Open Addressing-Linear Probing, Double Hashing.
Priority queues - Definition, ADT, Realizing a Priority Queue Using Heaps, Definition, Insertion, Deletion.

UNIT IV: (10 Lectures)
Search Trees - Binary Search Trees, Definition, ADT, Implementation, Operations-Searching, Insertion, Deletion.
AVL Trees, Definition, Height of AVL Tree, Operations, Insertion, Deletion and Searching.

UNIT V: (8 Lectures)
Search Trees - Introduction to Red-Black and Splay Trees, B-Trees, Height of B-Tree, Insertion, Deletion and Searching, Comparison of Search Trees.

Course Outcomes:

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

CO 1. Apply the knowledge of linked list, Stacks & Queues to solve the complex problems and analyze the performance.

CO 2. Assess the performance of Searching & Sorting Techniques and demonstrate the operations of Trees & Graphs to model nonlinear problems.

CO 3. Study essential concepts of ADT and evaluate the performance of various Hashing techniques.

CO 4. Illustrate the concepts of Priority Queues and Binary Search Trees for real world applications.

CO 5. Get the knowledge on necessity of height balanced trees and explore the rotations of AVL trees.

CO 6. Compare the operations of search trees like Red-Black, Splay and B-trees and their performance.

Mapping of COs to POs:

POs/ COs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	3	3	3	3	-	-	-	-	3	-	-
CO2	3	3	3	3	-	-	-	-	3	-	-
CO3	3	3	3	3	-	-	-	-	3	-	-
CO4	3	3	3	3	-	-	-	-	3	-	-
CO5	3	3	3	3	-	-	-	-	3	-	-
CO6	3	3	3	3	-	-	-	-	3	-	-

TEXT BOOKS:

1. Richard F, Gilberg, Forouzan, Data Structures, 2nd edition, Cengage
2. Samanta Debasis, Classic Data Structures, 2nd edition, PHI.
3. Mark Allen Weiss, Data structures and Algorithm Analysis in C, 2nd edition, Pearson Education. Ltd.

REFERENCE BOOKS:

1. Jean-Paul Tremblay Paul G. Sorenson, An Introduction to Data Structures with Applications, 2nd edition, Mc Graw Hill Higher Education.
2. Sartaj Sahni, Data Structures, Algorithms and Applications in java, 2/e, University Press.

SOFTWARE REQUIREMENTS AND ESTIMATION

Subject Code: PGCS1T1122	**L**	**T**	**P**	**C**
I Year / I Semester	**3**	**0**	**0**	**3**

Prerequisites: Familiarity with Software Engineering.

Course Objectives: This course demonstrates the good practices of requirements engineering, requirements elicitation, requirements modeling and software estimation.

Syllabus:

UNIT I **(6 Lectures)**

Software Requirements: What and Why: Essential Software requirement, Good practices for requirements engineering, Improving requirements processes, Software requirements and risk management.

UNIT II **(7 Lectures)**

Software Requirements Engineering: Requirements elicitation, requirements analysis documentation, review, elicitation techniques, analysis models, Software quality attributes, risk reduction through prototyping, setting requirements priorities, verifying requirements quality.

UNIT III **(10 Lectures)**

Software Requirements Management : Requirements management Principles and practices, Requirements attributes, Change Management Process, Requirements Traceability Matrix, Links in requirements chain.

Software Requirements Modeling: Use Case Modeling, Analysis Models, Dataflow diagram, state transition Diagram, class diagrams, Object analysis, Problem Frames.

UNIT IV **(10 Lectures)**

Software Estimation: Components of Software Estimations, Estimation methods, Problems associated with estimation, Key project factors that influence estimation.

Size Estimation: Two views of sizing, Function Point Analysis, Mark II FPA, Full Function Points, LOC Estimation, Conversion between size measures.

Effort, Schedule and Cost Estimation: What is Productivity? Estimation Factors, Approaches to Effort and Schedule Estimation, COCOMO II, Putnam Estimation Model, Algorithmic models, Cost Estimation.

UNIT V **(10 Lectures)**

Tools for Requirements Management and Estimation Requirements Management Tools: Benefits of using a requirements management tool, commercial requirements management tool, Rational Requisite pro, Caliber – RM, implementing requirements management automation.

Software Estimation Tools: Desirable features in software estimation tools, IFPUG, USC's COCOMO II, SLIM(Software Life Cycle Management) Tools.

Course Outcomes:

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

CO1: Understand and identify the importance of different types of Software Requirements.

CO2: Analyze software requirements engineering techniques and models.

CO3: Design software models by using software requirement management and modelling techniques.

CO4: Estimate the software in terms of size, cost, effort and schedule.

CO5: Identify the importance of tools required for requirement management, requirement estimation and software estimation.

Mapping of COs to POs:

POs/ COs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	3	-	-	-	-	-	-	-	-	-	-
CO2	3	3	-	3	-	-	-	-	2	2	-
CO3	3	3	3	-	3	-	-	-	2	2	3
CO4	3	3	-	3	3	-	-	-	-	-	-
CO5	3	-	-	-	-	-	-	-	-	-	-

TEXT BOOKS:

1. Rajesh Naik and Swapna Kishore, Software Requirements and Estimation, 1st edition, Tata Mc Graw Hill.

2. Karl E. Weigers, Software Requirements, 2nd edition Microsoft Press.

REFERENCE BOOKS:

1. Soren Lausen, Software Requirements Styles and Techniques, 1st edition, Addison-Wesley Professional.

2. Karl E.Weigers, Software Requirements Practical Techniques for gathering and managing requirements through the product development life cycle, 2nd edition, Microsoft Press.

SOFTWARE PROJECT AND PROCESS MANAGEMENT

Subject Code: PGCS1T1222	**L**	**T**	**P**	**C**
I Year / I Semester	**3**	**0**	**0**	**3**

Prerequisites: Basic concepts of software engineering.

Course Objectives: This course will help the students to develop their skills that will enable them to construct software of high quality that is reliable, reasonably easy to understand, modify and maintain.

Syllabus:

UNIT I: (8 Lectures)

Software Process Maturity Software maturity Framework: Principles of Software Process Change, Software Process Assessment, The Initial Process, The Repeatable Process, The Defined Process, The Managed Process, The Optimizing Process.

UNIT II: (9 Lectures)

Process Reference Models Capability Maturity Model (CMM): CMMI, PCMM, PSP, TSP, IDEAL, Process Definition Techniques.

Managing Software Projects: Project Management and the CMM, Project Management and CMMI, Project Management Process Framework.

UNIT III: (12 Lectures)

Software Project Management Renaissance: Conventional Software Management, Evolution of Software Economics, Improving Software Economics, The old way and the new way.

Project Planning: Software Life Cycle Models, Project Organizations and Responsibilities, Artifacts of the Project Management Process, Cost and Scheduling estimation, Establishing Project Environment, Risk Management, Quality Assurance and Configuration Management.

UNIT IV: (10 Lectures)

Project Tracking and Control: Defect Tracking, Issue Tracking, Status Reports, Milestone Analysis, Defect Analysis and Prevention Methods, Process monitoring and audit, Reviews, Inspections and Walkthroughs, Seven Core Metrics, Management indicators, Quality Indicators.

UNIT V: (7 Lectures)

Project Closure: Project Closure Analysis, Role of Closure Analysis in a project, Performing Closure Analysis, Closure Analysis Report.

Course Outcomes:

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

CO1: Demonstrate the importance of software maturity framework.

CO2: Get knowledge on different types of process models to suit the nature of a project and to manage software project.
CO3: Make use of the Software Project Management Renaissance
CO4: Understand the software project planning and risk management.
CO5: Determine the ways of defect tracking, control and status report.
CO6: Analyze the Project Closure and Closure Analysis Report.

Mapping of COs to POs:

POs/ COs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	3	-	-	-	-	-	-	3	-	-	-
CO2	3	-	-	-	-	-	-	-	3	3	-
CO3	3	-	-	-	-	-	-	-	-	-	-
CO4	3	-	-	-	-	-	-	-	-	-	-
CO5	3	-	-	-	2	-	-	-	-	3	3
CO6	3	3	-	-	2	-	-	-	-	-	-

TEXT BOOKS:
1. Watts S. Humphrey, Managing the Software Process, 1st edition Pearson Education.
2. Walker Royce, Software Project Management, 1st edition, Pearson Education.

REFERENCE BOOKS:
1. Watts S. Humphrey, An Introduction to the Team Software Process, 1st edition, Pearson Education.
2. Watts S. Humphrey, A Discipline to Software Engineering, 1st edition, Pearson Education.
3. Pankaj Jalote, Software Project Management in Practice, 1st edition, Pearson Education.

CLOUD COMPUTING

Subject Code: PGCS1T1322	**L**	**T**	**P**	**C**
I Year / I Semester	**3**	**0**	**0**	**3**

Prerequisites: Familiarity with Operating Systems, Computer Networks and Database Management Systems.

Course Objectives: The objective of this course is to provide students with the comprehensive and in-depth knowledge of Cloud Computing concepts, technologies, architecture and applications.

Syllabus:

UNIT I: (10 Lectures)

Introduction to Cloud Computing: Trends in Computing - Distributed Computing, Grid Computing, Cluster Computing, Utility Computing, Cloud Computing, Definition of Cloud Computing, Characteristics, Service Models, Deployment Models, Cloud Service Models Providers, Advantages and Disadvantages of Cloud Computing , Cloud-based Services & Applications.

UNIT II: (8 Lectures)

Cloud Concepts & Technologies: Virtualization and its types, Software Defined Networking, Network Function Virtualization(NFV).

Cloud Services: Compute Services, Storage Services, Database Services, Application Services.

UNIT III: (10 Lectures)

Cloud Application Design: Design Considerations for Cloud Applications, Reference Architectures for Cloud Applications, Cloud Application Design Methodologies: SOA, Cloud Component Model, and MVC, Data Storage Approaches.

Cloud Security: Cloud Security Architecture(CSA), Authentication, Authorization, Identity & Access Management, Data Security, Key Management.

UNIT IV: (7 Lectures)

Migrating into a Cloud: Broad Approaches to Migrating into the Cloud, The Seven-Step Model of Migration into a Cloud, Migration Risks and mitigation, Phases of Migrating to Cloud, benefits and risks of Migrating to Cloud.

UNIT V: (9 Lectures)

SLA Management in Cloud Computing: Service Level Agreements(SLA), Considerations for SLA, SLA Requirements, Types of SLA, Life Cycle of SLA, SLA

Management in Cloud. **Case Study:** Amazon AWS: EC2, Amazon Simple DB, Amazon S3, Amazon Cloud Front and Amazon SQS.

Course Outcomes:

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

CO 1. Illustrate key technologies, strengths, and limitations of cloud computing and the possible applications for state-of-the-art cloud computing.

CO 2. Choose the appropriate methodologies and considerations for Cloud application design.

CO 3. Interpret the core issues of Cloud Computing such as security, Privacy and Interoperability.

Mapping of COs to POs:

POs/ COs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	3	2	-	-	-	-	-	-	-	-	-
CO2	3	3	-	-	-	-	-	-	-	-	3
CO3	3	3	-	3	-	-	-	-	3	3	3

TEXT BOOKS:

1. Arshdeep Bahga, Vijay Madisetti, Cloud Computing : A Hands-on Approach, Universities Press.
2. Rajkumar Buyya, James Brogerg, Andrzej Goscinski, Cloud Computing : Principles and Paradigms, WILEY Publication.

REFERENCE BOOKS:

1. Michael Miller, "Cloud Computing – Web Based Applications That Change the way you Work and Collaborate Online", Pearson Education.
2. Anthony T. Velte Toby J. Velte, Robert Elsenpeter, "Cloud Computing : A Practical Approach", McGraw-Hill.
3 Arshdeep B, Vijay M, "Cloud computing: A Hands-on Approach", Universities Press.

ARTIFICIAL INTELLIGENCE
(ELECTIVE–I)

Subject Code: PGCS1T0522	**L**	**T**	**P**	**C**
I Year / I Semester	**3**	**0**	**0**	**3**

Prerequisites: Familiarity with Discrete Mathematics, Linear Algebra and Probability.

Course Objectives: The objective of the course is to present an overview of artificial intelligence principles and approaches.

Syllabus:

UNIT I: (7 Lectures)

Introduction to artificial intelligence: Introduction, history, intelligent systems, foundations of AI, applications, tic-tac-toe game playing, development of AI languages, current trends in AI. **Problem solving: state-space search and control strategies:** Introduction, general problem solving, characteristics of problem.

UNIT II: (9 Lectures)

Search Strategies: exhaustive searches, heuristic search techniques, iterative-deepening a*, constraint satisfaction. **Problem reduction and game playing:** Introduction, problem reduction, game playing, alpha-beta pruning, two-player perfect information games.

UNIT III: (10 Lectures)

Logic concepts: Introduction, propositional calculus, proportional logic, natural deduction system, axiomatic system, semantic tableau system in proportional logic, resolution refutation in proportional logic, predicate logic.

Knowledge representation: Introduction, approaches to knowledge representation, knowledge representation using semantic network, extended semantic networks for KR, knowledge representation using frames.

UNIT IV: (10 Lectures)

Advanced knowledge representation techniques: Introduction, conceptual dependency theory, script structure, cyc theory.

Expert system and applications: Introduction phases in building expert systems, expert system versus traditional systems, rule-based expert systems blackboard systems truth maintenance systems, application of expert systems, list of shells and tools.

UNIT V: (8 Lectures)

Uncertainty measure: probability theory: Introduction, probability theory, Bayesian belief networks, certainty factor theory, dempster-shafer theory. **Fuzzy**

sets and fuzzy logic: Introduction, fuzzy sets, fuzzy set operations, types of membership functions, multi valued logic, fuzzy logic, linguistic variables and hedges, fuzzy propositions, inference rules for fuzzy propositions, fuzzy systems.

Course Outcomes:

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

- **CO 1** Summarize and formulate appropriate logic concepts and AI methods for solving a problem.
- **CO 2** Applying various searching, game playing, and knowledge representation techniques to solve the real world problems.
- **CO 3** Analyze different expert systems and its applications.
- **CO 4** Explain the concepts of probability theory, fuzzy sets and fuzzy logic for uncertainty measure.

Mapping of COs to POs:

POs/ COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11
CO1	3	3	3	3	-	-	-	-	-	-	-
CO2	3	3	3	-	-	-	-	-	3	-	-
CO3	3	3	-	3	-	-	-	-	3	-	-
CO4	3	3	-	-	-	-	-	-	-	-	-

TEXT BOOKS:

1. Saroj Kaushik, Artificial Intelligence, CENGAGE Learning.
2. Stuart Russel, Peter Norvig, Artificial intelligence, A modern Approach, 2nd ed, PEA.
3. Rich, Kevin Knight, Shiv Shankar B Nair, Artificial Intelligence, 3rd ed, TMH.
4. Patterson, Introduction to Artificial Intelligence, PHI.

REFERENCE BOOKS:

1. George F Lugar ,Artificial intelligence, structures and Strategies for Complex problem solving, 5th ed, PEA.
2. Ertel, Wolf Gang, Introduction to Artificial Intelligence, Springer.
3. Nils J Nilsson, A new Synthesis Artificial Intelligence, Elsevier.

DATA SCIENCE WITH R

Subject Code: PGCS1T1422	**L**	**T**	**P**	**C**
I Year / I Semester	**3**	**0**	**0**	**3**

Prerequisites: Familiarity with the concepts of databases and statistics.

Course Objectives: The objective of this course is to provide insights to learner's about data science process and exploration in real-time perspectives. This course also focuses on various concepts such as types of learning, processes, techniques and models concerned to data science by using R programming language.

Syllabus:

UNIT I: (6 Lectures)

Introduction to Data Science: Data Science, Data Science Applications, Data Science related fields, Relationship between Data Science and Information Science, Computational Thinking, Skills for Data Science, Tools for Data Science, Issues of Ethics, Bias and Privacy in Data Science.

UNIT II: (9 Lectures)

Introduction to R: How to run R, A first R Session, Functions, Important R Data Structures: Vectors, Character strings, Matrices, Lists, Data Frames and Classes, Getting Help.

R Programming Structures: Control Statements, Arithmetic and Boolean Operators and values, Default Values for Arguments, Return Values, Functions are Objects, Environment and Scope Issues, Tools for Composing Function Code.

UNIT III: (9 Lectures)

Doing Math in R: Math Functions, Functions for Statistical Distributions, Sorting, Set Operations, Simulation Programming in R.

Input/Output in R: Accessing the Keyboard and Monitor, Reading and Writing Files.

Graphics in R: Creating Graphs, Customizing Graphs, Saving Graphs to Files, Creating Three-Dimensional Plots.

UNIT IV: (10 Lectures)

Nature of Data & Pre-processing: Introduction, Data Types, Data Collections, Data Pre-processing.

Data Science Techniques: Introduction, Data Analysis and Data Analytics, Descriptive Analysis, Diagnostic Analytics, Predictive Analytics, Prescriptive Analytics, Exploratory Analysis, Mechanistic Analysis.

Association Rules: Overview, Apriori Algorithm, Evaluation of Candidate Rules, Applications of Association Rules, Validation and Testing.

UNIT V: (10 Lectures)

Regression: Linear Regression, Logistic Regression, Reasons to Choose and Cautions.

Classification: Decision Trees - Overview, Decision Tree Algorithms, Evaluating a Decision Tree, Decision Trees in R, Naive Bayes - Bayes' Theorem, Naive Bayes Classifier, Smoothing, Diagnostics and Naive Bayes in R, Diagnostics of Classifiers.
Clustering: Overview of Clustering, K-means - Use Cases, Overview of the Method, Determining the Number of Clusters, Diagnostics.

Course Outcomes:
Upon completion of this course, the students will be able to:
CO-1: Relate the need for the data science process and its applications in real-time perspectives.
CO-2: Demonstrate data exploration, manipulation and visualization by using various functions and packages in the R programming language
CO-3: Interpret data processing and various types of analytics on data streams
CO-4: Analyse the application of data mining algorithms on different data sources to extract data insights.

Mapping of COs to POs:

POs/ COs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	3	3	3	3	3	-	-	-	2	-	-
CO2	3	3	3	3	3	-	-	-	2	-	-
CO3	3	3	3	3	3	-	-	-	2	-	-
CO4	3	3	3	3	3	-	-	-	2	-	-

TEXT BOOKS:
1. Chirag Shah, A Hands-On Introduction to Data Science, Cambridge University Press.
2. Norman Matloff, The Art of R Programming, No Starch Press.
3. David Dietrich, Barry Heller and Beibei Yang, Data Science and Big Data Analytics, Wiley.

REFERENCE BOOKS:
1. Rob Kabacoff, R in Action - Data Analysis and Graphics with R, Manning Publications.
2. Vijay Kotu, Bala Deshpande, "Data Science: Concepts and Practice", Second Edition, Elsevier Publications.
3. C. O'Neil and R. Schutt, Doing Data Science: Straight Talk from the Frontline, O'Reilly.
4. Joel Grus, Data Science from Scratch, O'Reilly.
5. Jared P. Lander, R for Everyone: Advanced Analytics and Graphics, Addison-Wesley Data& Analytics Series.

INTERNET OF THINGS
(Elective-I)

Subject Code: PGCS1T0722	**L**	**T**	**P**	**C**
I Year / I Semester	**3**	**0**	**0**	**3**

Prerequisites: Familiarity with Digital Electronics.

Course Objectives: Students will be explored to the interconnection and integration of the physical world and the cyber space. They are also able to design & develop IOT enabled Devices.

Syllabus:

UNIT I: (8 Lectures)
Introduction to Internet of Things: Introduction: Definition of IoT, Fundamental Characteristics of IoT, Design Considerations for IoT Applications.

UNIT II: (8 Lectures)
Basic layered architecture for IoT: Device Layer, Network Layer, Service and Application Support Layer, Application Layer, Structure of IoT, Logical design of IoT.

UNIT III: (8 Lectures)
Key enabling Technologies: Platforms: Hardware, SoC, Sensors, Bluetooth, BT-LE, iBeacon. **Protocols:** Identification and Tracking Technologies: RFID, NFC, Zigbee and GPS Communication Technologies: Wireless Networks, WSN, 3G, LTE, IPv6.
IoT protocols: HTTP, CoAp, Websocket, MQTT, XMPP, DDS, AMQP, SDN and NFV for IoT.

UNIT IV: (10 Lectures)
Services and attributes for IoT: Gateway, Raspberry Pi, Arduino, Cloud Computing and IoT, Big-Data Analytics and Visualization, Dependability, Security, Localization, Maintainability.
Internet of Things in Practice: IoT levels and Deployment templates, IoT for Smart Cities, IoT for Agriculture, IoT for Traffic Management and Transportation, IoT in the Home, IoT in Retail, IoT in Healthcare, IoT in Sports.

UNIT V: (8 Lectures)
Challenges and Future Trends: Research challenges: Technical Challenges, Standardization, Information Security and Privacy Protection, Research Trends.

Course Outcomes:
Upon completion of this course, the students will be able to:
CO1: Learn the fundamentals of IOT.
CO2: Interpret the Architectural layered view of the IOT.

CO3: Analyze the basic protocols of IOT.
CO4: Design IOT applications in various domains and able to analyze the performance.
CO5: Integrate IOT Applications with Embedded Platform to enhance the future .

Mapping of COs to POs:

POs/ Cos	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	3	3	-	-	-	-	-	-	-	-	-
CO2	3	3	-	-	-	-	-	-	-	-	-
CO3	3	3	-	-	-	-	-	-	-	-	-
CO4	3	3	3	3	3	2	-	2	2	2	2
CO5	3	3	3	3	3	3	2	2	3	2	3

TEXT BOOKS:
1. Arshadeep Bhaga, Vijay madisetti, IoT A hands on approach,University Press.
2. Ovidiu Vermesan, Peter Friess, Internet of Things – From Research and Innovation to Market Deployment, River Publishers.
3. Ovidiu Vermesan, Peter Friess, Internet of Things – Converging Technologies for Smart Environments and Integrated Ecosystems, River Publishers.

REFERENCE BOOKS:
1. Olivier Hersent, David Boswarthick, Omar Elloumi, The Internet of Things: Key Applications and Protocols, Wiley.
2. Ovidiu Vermesan, Peter Friess, Building the Hyperconnected Society, River Publishers.

DEVOPS
(ELECTIVE-I)

Subject Code: PGCS1T1522	**L**	**T**	**P**	**C**
I Year / I Semester	**3**	**0**	**0**	**3**

Prerequisites: Good exposure to Software Engineering concepts and Software Development Methodologies.

Course Objectives:
To get an expertise on the culture of DevOps in Software Development Methodologies for finding ways to adapt and innovate social structure, culture and technology together in order to work more effectively in the Enterprises.

Syllabus:

UNIT I: (8 Lectures)
Introduction to DevOps: What is DevOps, A History of DevOps, Fundamental Terminology and Concepts – Software Development Methodologies, Operations Methodologies, Systems Methodologies, Development Release and Deployment Concepts, Infrastructure Concepts, Cultural Concepts. DevOps Misconceptions and Anti-Patterns, the Four Pillars of Effective DevOps.

UNIT II: (8 Lectures)
Collaboration: Defining Collaboration, Individual Differences and Backgrounds, Opportunities for Competitive Advantage, Mentorship, Introducing Mindsets, Mindsets and Learning Organizations, The Role of Feedback, Reviews and Rankings, Communication and Conflict Resolution Styles, Empathy and Trust, Humane Staffing and Resources, Misconceptions and Troubleshooting of Collaboration.

UNIT III: (12 Lectures)
Affinity: What Makes a Team, Teams and Organizational Structure, Finding Common Ground Between Teams, Benefits of Improved Affinity, Requirements for Affinity, Measuring Affinity, Misconceptions and Troubleshooting of Affinity.
Tools: Software Development, Automation, Monitoring, Evolution of the Ecosystem,The Value of Tools to People, What Are Tools?, The Right Tools for Real Problems, Embracing Open Source, Standardization of Tools, Consistent Processes for Tool Analysis, Exceptions to Standardization, Irrelevance of Tools, The Impacts of Tools on Culture, Selection of Tools, Auditing Your Tool Ecosystem, Elimination of Tools, Misconceptions and Troubleshooting of Tools.

UNIT IV: (8 Lectures)
Scaling: Understanding Scaling, Considering Enterprise DevOps, Organizational Structure, Team Flexibility, Organizational Lifecycle, Complexity and Change, Scaling for Teams, Team Scaling and Growth Strategies, Scaling for Organizations, Misconceptions and Troubleshooting of Scaling.

UNIT V: (6 Lectures)

DevOps Practices: Implementing CI/CD and continuous deployment, Understanding IaC practices, DevOps Best Practices: Automating everything, Choosing the right tool,Writing all your configuration in code, Designing the system architecture, Building a good CI/CD pipeline, Integrating tests, Applying security with DevSecOps, Monitoring your system, Evolving project management.

Course Outcomes:

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

CO 1. Make use the Influence of DevOps on Software Development Methodologies along with its Misconceptions and Anti-Patterns.

CO 2. Illustrate the Methodologies of Four Pillars of DevOps and Troubleshoot the common problems that can arise in the effective DevOps.

CO 3. Inference the culture of DevOps to the Enterprises for achieving agility and innovation in its business units.

Mapping of COs to POs:

POs/ COs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	3	3	3	-	-	-	-	-	-	-	3
CO2	3	3	3	-	-	-	-	-	-	-	3
CO3	3	3	-	-	-	-	-	-	-	-	-

TEXT BOOKS:

1. Jennifer Davis, RynDaniels, Effective DevOps: Building a Culture of Collaboration, Affinity, and Tooling at Scale, O'Reilly.

2. Mikael Krief, Learning DevOps, Packt Publications.

REFERENCE BOOKS:

1. Verona, Joakim. Practical DevOps. Packt Publishing Ltd.

2. By Jez Humble and David Farley, Continuous Delivery: Reliable Software Releases through Build, Test and Deployment Automation, Addison-Wesley Professional

3. Mandi Walls,Building a DevOps Culture, O'Reilly publications.

4. Sanjeev Sharma, "The DevOps Adoption Playbook – A Guide to Adopting DevOps in a Multi-Speed IT Enterprise", Wiley Publications.

ADVANCED DATA STRUCTURES LAB

Subject Code: PGCS1P0922	**L**	**T**	**P**	**C**
I Year / I Semester	**0**	**0**	**3**	**1.5**

Prerequisites:
Students should have the familiarity with the concepts of Programming along with basic constructs of C language to implement the data structure algorithms.

Course objectives:
This course is concerned with the design and analysis of efficient algorithms of linear and non- linear data structures, focusing principally on algorithms for combinatorial optimization problems. The course is intended to Strengthen the ability of the students to identify and implement the suitable data structure for the given real-world problem and gain knowledge in practical applications of data structures.

List of Experiments:

1. To implement the operations of Singly Linked List and Circular Linked List.

2. To implement the operations of Doubly Linked List.

3. To perform various operations of Stack and Queue using Linked List.

4. To implement operations on Graph.
    a. Vertex insertions
    b. Vertex deletions
    c. Finding vertex
    d. Edge addition and deletion

5. a. Write a program to implement BFS traversal on given Graph.
    b. Write a program to implement DFS traversal on given Graph.

6. To implement functions of Dictionary using hashing.
    a. Division Method
    b. Multiplication Method
    c. Universal Hashing

7. Write a program to perform
    a. Linear Probing
    b. Random Probing
    c. Double Hashing

8. Write a program to implement Priority Queue using
    a. Max Heap
    b. Min Heap

9. To perform various operations of Binary Search Trees.

10. To perform various operations of AVL Trees.

Course Outcomes:

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

CO 1. Proficient in writing the programs to implement the operations of SLL, CLL and DLL.
CO 2. Design and develop operations of Stack & Queue using Linked List.
CO 3. Generate code for operations of Graphs and Traversals for real time problems.
CO 4. Enhance the expertise in practice knowledge of Hash Functions and Collision Resolution Methods.
CO 5. Model the contextual solution to a given problem using Search Trees.

Mapping of COs to POs:

POs/ COs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	3	3	3	3	-	3	-	3	-	-	3
CO2	3	3	3	3	-	3	-	3	-	-	3
CO3	3	3	3	3	-	3	-	3	-	-	3
CO4	3	3	3	3	-	3	-	3	3	-	3
CO5	3	3	3	3	-	3	-	3	-	-	3

TEXT BOOKS:

1. Richard F, Gilberg, Forouzan, Data Structures, 2nd edition, Cengage
2. Samanta Debasis, Classic Data Structures, 2nd edition, PHI.
3. Mark Allen Weiss, Data structures and Algorithm Analysis in C, 2nd edition, Pearson Education. Ltd.

REFERENCE BOOKS:

1. Jean-Paul Tremblay Paul G. Sorenson, An Introduction to Data Structures with Applications, 2nd edition, Mc Graw Hill Higher Education
2. Sartaj Sahni, Data Structures, Algorithms and Applications in java, 2/e, University Press.

CLOUD COMPUTING LAB

Subject Code: PGCS1P1622	**L**	**T**	**P**	**C**
I Year / I Semester	**0**	**0**	**3**	**1.5**

Prerequisites: Familiarity with Computer Networks, Virtualization and Scheduling algorithms.

Course Objectives: This course is aimed to implement the basic concepts of cloud computing like virtualization, migration, services and deployment models.

List of Experiments:

1. Implement full virtualization concept in Xen hypervisor.
2. Implement paravirtualization concept in KVM.
3. Configure Infrastructure as a Service using Openstack.
4. Configure Storage as a Service using Owncloud.
5. Migrate a single instance to another compute host using OpenStack.
6. Configure administrative features of Cloud Management.
7. Configure administrative features of User Management.
8. Create an instance in AWS.
9. Create an S3 bucket using AWS S3 and set the necessary ACL permissions for that bucket.
10. Develop a chat bot using AWS lex service.

Course Outcomes:

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

CO 1 Explain how to configure Cloud Services.
CO 2 Implement Virtualization Concepts.
CO 3 Deploy the instances in public clouds.
CO 4 Analyse accountability of cloud resources.

Mapping of COs to POs:

POs/ COs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	3	-	-	-	-	-	-	3	-	-	-
CO2	3	-	3	-	3	3	3	3	3	3	3
CO3	3	-	3	-	3	3	3	3	3	3	3
CO4	3	3	-	3	3	-	-	-	-	-	-

TEXT BOOKS:

1. Arshdeep Bahga, Vijay Madisetti, Cloud Computing: A Hands-on Approach, Universities Press.
2. Rajkumar Buyya, James Brogerg, Andrzej Goscinski, Cloud Computing : Principles and Paradigms, WILEY Publication.

REFERENCE BOOKS:

1. Michael Miller, "Cloud Computing – Web Based Applications That Change the way you Work and Collaborate Online", Pearson Education.
2. Anthony T. Velte Toby J. Velte, Robert Elsenpeter, "Cloud Computing : A Practical Approach", McGraw-Hill.

I Year
II Semester

FULL STACK DEVELOPMENT

Subject Code: PGCS2T0322	**L**	**T**	**P**	**C**
I Year / II Semester	**3**	**0**	**0**	**3**

Prerequisites: The students must have knowledge on the concepts of HTML, CSS and JavaScript.

Course Objectives:
Full Stack Development Course covers complete breath of technologies & applications that are extensively used in the industry. This course not only make the students gain expertise in both front end & backend programming applications but also help them to get familiar with the latest web development technologies & also complete web development life cycle.

Syllabus:

UNIT I: (8 Lectures)
React Basics: Introduction, Nesting elements, Creating component classes, Working with properties, Introduction to JSX, States, Component lifecycle events.

UNIT II: (9 Lectures)
React Advanced Concepts: Handling events, Working with forms, Scaling React components, React routing, Working with data using Redux, Unit testing React with Jest.

UNIT III: (10 Lectures)
Node.js: Introduction, Modules and npm, Node's Programming Model, Core Modules, Building the Node Server, Node's Debugger, node-inspector, Testing Node.

MongoDB: History of MongoDB, Installing MongoDB Locally, Cloud Hosting, MongoDB Shell, Inserting New Data, Retrieving Data, Updating Data, Deleting Data, Deleting Collections, Deleting Databases, Interacting with MongoDB Using Mongoose, Alternatives to MongoDB.

UNIT IV: (7 Lectures)
Express.js : Building Blocks of Express, Router, Middleware, Routes, Generating an Express App, Jade, Architecture of an Express Application.

UNIT V: (8 Lectures)
AngularJS: Single-page Applications, SPA Frameworks, Model-View-Controller Architecture, Getting Angular, Data Binding, Angular Directives, Controllers, Client-side Routing, Testing Angular.

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

CO 1. Understand the principles, working methodologies and operations of front end and back end programming applications.
CO 2. Demonstrate the design methodology of MEAN architecture frame works to support real time interactions.
CO 3. Apply the techniques of modern web methodologies for formulating solutions to real world problems.
CO 4. Analyze and Integrate all the components of for developing robust and dynamic web applications.

Mapping of COs to POs:

POs/ COs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	3	-	-	-	-	-	-	-	-	-	-
CO2	3	3	-	-	-	-	-	3	-	-	-
CO3	3	-	3	-	3	3	-	-	-	-	-
CO4	3	3	3	-	-	3	-	-	-	-	-

TEXT BOOKS:

1. Azat Mardan, React Quickly, Manning Publications Co.
2. Colin J. Ihrig and Adam Bretz, Full Stack JavaScript Development with MEAN, SitePoint Pty Ltd.

REFERENCE BOOKS:

1. Kirupa Chinnathambi, Learning React, Pearson Education Inc.
2. Cássio de Sousa Antonio, Pro React, Apress.
3. Vasan Subramanian, Pro MERN Stack, Apress Publications.
4. Simon Holmes, Clive Harber, Getting Mean with Mongo, Express, Angular and Node, Manning Publications Co.
5. Amos Q. Haviv, MEAN Web Development, Packt Publishing.

SOFTWARE ARCHITECTURE AND DESIGN PATTERNS

Subject Code: PGCS2T1422	**L**	**T**	**P**	**C**
I Year / II Semester	**3**	**0**	**0**	**3**

Prerequisites: Familiarity with Software Engineering and Object Oriented Analysis & Design.

Course Objectives: This course focuses on design, creation and analysis of Software Architectures. It also focuses on the Design Patterns and its community.

Syllabus:

UNIT I: (8 Lectures)
Envisioning Architecture: The Architecture Business Cycle, Software Processes and the Architecture Business Cycle, Software Architecture, Good Architecture, Architectural Patterns, Reference Models and Reference Architectures, Importance of Software Architecture, Architectural Structures and Views.

UNIT II: (9 Lectures)
Creating an Architecture: Quality Attributes, Moving from quality to architecture, Architectural styles and patterns, Achieving qualities, Designing the Architecture, Documenting the architecture, Reconstructing Software Architecture, shared information systems.

UNIT III: (10 Lectures)
Analyzing Software Architecture: Analyzing development qualities at the architectural level, SAAM, ATAM, CBAM, Architecture Reviews.
Moving from Architecture to Systems: Software Product Lines, Building systems from off the shelf components, Reuse of Architectural assets within an organization.

UNIT IV: (8 Lectures)
Patterns: What is pattern? Pattern categories, Pattern Description, Patterns and Software Architecture, Pattern Systems, Classification, Selection.

UNIT V: (9 Lectures)
Design Patterns Catalog: Creational Pattern, Structural Pattern, Behavioral Patterns, Pattern Community, Designing a document editor.

Course Outcomes:
Upon completion of this course, the students will be able to:
CO1: Argue the importance and role of software architecture in large-scale software systems.
CO2: Analyze, design and evaluate a system's architecture.
CO3: Construct the architecture of software and achieve the quality.
CO4: Recognize and understand several design patterns.

CO5: Select and use appropriate software design patterns.

Mapping of COs to POs:

POs/ COs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	3	3	-	-	-	-	-	-	-	-	-
CO2	3	3	-	3	3	-	-	-	3	-	-
CO3	3	3	3	-	3	-	-	2	-	-	-
CO4	3	3	-	-	3	-	-	-	3	-	3
CO5	3	3	-	-	3	-	-	-	3	-	3

TEXT BOOKS:

1. Len Bass, Paul Clements & Rick Kazman, Software Architecture in Practice, second edition, Pearson Education.
2. Erich Gamma, Design Patterns, Pearson Education.

REFERENCE BOOKS:

1. Luke Hohmann, Beyond Software architecture, Addison Wesley.
2. David M.Dikel, David Kane and James R.Wilson, Software architecture, Prentice Hall.
3. David Budgen, Software Design, second edition, Pearson education.
4. Eric Freeman & Elisabeth Freeman, Head First Design patterns, O'REILLY.
5. Steven John Metsker & William C. Wake, Design Patterns in Java, Pearson education.
6. F.Buschmann &others,Pattern Oriented Software Architecture, John Wiley & Sons.

SOFTWARE QUALITY ASSURANCE AND TESTING

Subject Code: PGCS2T1522	**L**	**T**	**P**	**C**
I Year / II Semester	**3**	**0**	**0**	**3**

Prerequisites: Basic Knowledge on Software Engineering.

Course Objectives: The main objective of this course is to understand the software quality assurance and software testing are the fundamental components of software life cycle. And impart knowledge on the activities of software quality assurance with tools and techniques to develop software in efficient way in terms of cost, effort and quality.

Syllabus:

UNIT I: (9 Lectures)

Software quality assurance Framework and Standards SQA Frame work: What is Quality? Software Quality Assurance, Components of Software quality Assurance.

Software Quality Assurance Plan: Steps to develop and implement a Software quality Assurance Plan.

Standards: ISO9000, CMM, CMMI, PCMM, Malcom Balridge, 3 Sigma, 6 Sigma.

UNIT II: (10 Lectures)

Software Quality Assurance Metrics and Measurement Software Quality Assurance Metrics: Product Quality metrics, In- Process Quality metrics, Metrics for Software Maintenance, Examples of Metric Programs, Software quality indicators, Fundamentals in Measurement Theory.

Building Software Testing Environment: Writing Policy for software testing, Economics of testing, Building a structured approach to software testing.

UNIT III: (10 Lectures)

Software Testing process: Defects Hard to find, Functional and structured testing, Workbench concept, customizing the software testing process, testing tactics check list.

Software Testing Techniques: Black-Box testing, Boundary value analysis, Bottom-up, Branch Coverage, Cause- Effect graphing, CRUD, Database, exception, Gray box, Histogram, Inspections, JADs, Pareto Analysis, prototyping, random Testing, Risk based Testing, Regression Testing, Structured Walkthrough, Thread testing, Performance Testing, White Box Testing.

UNIT IV: (9 Lectures)

Software Testing Tools: Taxonomy of Testing tools, Methodology to evaluate automated testing tools, Load Runner, Win Runner and Rational Testing Tools, Java testing Tools, JMeter, JUNIT and Cactus.

Testing Process: Advantages of following a process, Cost of computer testing, Seven step software Testing Process, Define the scope of testing, Developing the test plan, Verification Testing. Validation Testing, Analyzing and reporting test results, Acceptance and operational Testing, Post Implementation Analysis.

UNIT V: (8 Lectures)

Testing Specialized Systems and Applications: Testing Client/Server System, Testing COTS and Contracted Software, Testing security, Testing Data Warehouse.

Course Outcomes:

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

CO 1. Understand the fundamental standards of SQA framework and quality assurance plan.

CO 2. Get the familiarity with software quality assurance metrics and distinguish product quality metrics and process quality metrics.

CO 3. Study essential testing environment and outline various software testing tools.

CO 4. Explore Testing techniques for various specialized systems and its applications.

Mapping of COs to POs:

POs/ COs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	3	-	-	-	-	-	-	-	-	-	-
CO2	3	-	-	-	2	-	-	-	3	-	-
CO3	3	-	-	-	3	-	-	-	3	-	-
CO4	3	3	-	3	2	-	-	-	3	-	3

TEXT BOOKS:

1. William E.Perry: Effective Methods for Software Testing, 3 rd Edition, Wiley Publication.
2. Mordechai Ben-Menachem, Garry S. Marliss: Software Quality, 1st Edition, Thomson Learning Publication.

REFERENCE BOOKS:

1. Kshirasagar Naik, PriyadarshiTripathy, Software Testing and Quality Assurance, Wiley.

MACHINE LEARNING
(ELECTIVE–II)

Subject Code: PGCS2T0422	**L**	**T**	**P**	**C**
I Year / II Semester	**3**	**0**	**0**	**3**

Prerequisites: Basic programming skills, algorithm design and basics of probability & statistics.

Course Objectives:

The main objective of this course is to understand the key algorithms and theory that form the foundation of machine learning and computational intelligence, as well as to identify and apply the appropriate machine learning technique for classification, pattern recognition, optimization and decision problems.

Syllabus:

UNIT I: (8 Lectures)

Introduction : Well-posed learning problems, Designing a learning system, Perspectives and issues in machine learning. Concept learning and the general to specific ordering – Introduction, A concept learning task, Concept learning as search, Find- S: finding a maximally specific hypothesis, Version spaces and the candidate elimination algorithm, Remarks on version spaces and candidate elimination, Inductive bias.

UNIT II: (8 Lectures)

Linear Regression & Logistic Regression: Predicting numeric values: regression – Finding the best fit lines with linear regression, Locally weighted linear regression, Shrinking Coefficients, The bias / Variance tradeoff. Logistic Regression: Classification with logistic regression and the sigmoid function, Using optimization to find the best regression coefficients.

UNIT III: (12 Lectures)

Decision Tree Learning: Decision tree representation, appropriate problems for decision tree learning, basic decision tree learning algorithm, and hypothesis space search in decision tree learning, inductive bias in decision tree learning, and issues in decision tree learning.

Artificial Neural Networks: Introduction, Neural network representation, Appropriate problems for neural network learning, Perceptions, Multilayer networks and the back propagation algorithm, Remarks on the back propagation algorithm, An illustrative example face recognition, Advanced topics in artificial neural networks.

UNIT IV: (8 Lectures)

Support Vector Machines & Dimensionality Reduction Techniques: Separating data with the maximum margin, finding the maximum margin, efficient optimization with SMO algorithm, speeding up optimization with full platt SMO, Using Kernels for more Complex data. Dimensionality Reduction techniques: Principal Component analysis, Example.

UNIT V: (8 Lectures)

Instance-Based Learning - Introduction, k -Nearest Neighbor Learning, Locally Weighted Regression, Radial Basis Functions, Case-Based Reasoning, Remarks on Lazy and Eager Learning.

Course Outcomes:

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

CO 1. Summarize the concepts and methods related to machine learning such as classification, regression, clustering, bias/variance, kernel functions, and optimization.

CO 2. Predict the expected outcome of the problem based on the training data by applying specific machine learning algorithm.

CO 3. Implement and compare the relevant algorithms using performance metrics.

CO 4. Design and build the model using various machine learning algorithms in a range of real-world applications.

Mapping of COs to POs:

POs/ COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11
CO1	3	3	-	-	-	-	-	-	-	-	-
CO2	3	3	-	3	-	-	-	-	-	-	-
CO3	3	3	3	3	-	-	-	-	-	-	-
CO4	3	-	3	-	-	-	-	-	-	-	-

TEXT BOOKS:

1. Tom M. Mitchell, Machine Learning, MGH.
2. Peter Harington, Machine Learning in Action, Cengage.
3. Sebastian Raschka, Python Machine Learning, Packt Publishing.

REFERENCE BOOKS:

1. Ethem Alpaydin, Introduction to Machine Learning, PHI.
2. Drew Conway & John Miles Wine, Machine Learing for Hackers, O'Reilly Media.

NOSQL DATABASES
(ELECTIVE–II)

Subject Code: PGCS2T1622	**L**	**T**	**P**	**C**
I Year / II Semester	**3**	**0**	**0**	**3**

Prerequisites: The students should have knowledge on Database Management Systems.

Course Objectives:
The course is intended to familiarize with various NoSQL Databases to handle structured, semi-structured and unstructured data.

Syllabus:

UNIT I: (8 Lectures)
Introduction: Early Database Management Systems, Relational Database Revolution, Motivations for NoSQL Databases, Four Types of NoSQL Databases - Key-Value Databases, Document Databases, Column Family Databases, Graph Databases.

UNIT II: (9 Lectures)
Key-Value Databases: From Arrays to Key-Value Databases, Essential Features of Key-Value Databases, Keys: More Than Meaningless Identifiers, Values: Storing Just About Any Data You Want, Key-Value Database Data Modeling Terms, Key-Value Architecture Terms, Key-Value Implementation Terms, Key Design and Partitioning, Designing Structured Values, Limitations of Key-Value Databases.

UNIT III: (12 Lectures)
Document Databases: What is a Document?, Avoid Explicit Schema Definitions, Basic Operations on Document Databases, Document and Collection Terms, Types of Partitions, Data Modeling and Query Processing, Designing for Document Databases.
Column Family Databases: Column Family Database Features, Architectures Used in Column Family Databases, When to Use Column Family Databases, Basic Components of Column Family Databases, Implementing Column Family Databases, Internal Structures and Configuration Parameters, Clusters and Partitions, Processes and Protocols, Designing for Column Family Databases.

UNIT IV: (9 Lectures)
Graph Databases: What Is a Graph?, Graphs and Network Modeling, Advantages of Graph Databases, Elements of Graphs, Operations on Graphs, Properties of Graphs and Nodes, Types of Graphs, Designing for Graph Databases.

UNIT V: (6 Lectures)
Guidelines for Selecting a NoSQL Database: Selecting Key-Value Databases, Selecting Document Databases, Selecting Column Family Databases, Selecting Graph Databases, Using NoSQL and Relational Databases together, List of NoSQL Databases.

Course Outcomes:

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

CO 1. Understand the need of NoSQL Databases and compare different NoSQL Databases such as Key-Value, Document, Column Family and Graph Databases.

CO 2. Demonstrate competency in designing Key-Value and Document databases.

CO 3. Demonstrate competency in designing Column Family and Graph databases.

CO 4. Choose NoSQL databases for a specific application.

Mapping of COs to POs:

POs/ COs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	3	2	-	-	-	-	-	-	-	-	-
CO2	3	3	3	3	-	-	-	3	-	-	3
CO3	3	3	3	3	-	-	-	3	-	-	3
CO4	3	3	-	-	-	-	-	-	-	-	3

TEXT BOOKS:

1. Dan Sullivan, "NoSQL for Mere Mortals", Addison Wesley.
2. Adam Fowler, NoSQL For Dummies, John Wiley & Sons Inc.

REFERENCE BOOKS:

1. Guy Harrison, Next-Generation Databases, Apress.
2. Shashank Tiwari, Professional NoSQL, Wrox Press
3. Pramod J Sadalage, Martin Fowler, "NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence", Addison Wesley.
4. Dan McCreary and Ann Kelly, Making Sense of NoSQL, Manning Publications
5. Eric Redmond, Jim R Wilson, "Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement", O'Reilly.
6. Gaurav Vaish, Getting Started with NoSQL, Packt Publishing.

CYBER SECURITY
(ELECTIVE-II)

Subject Code: PGCS2T1022	**L**	**T**	**P**	**C**
I Year / II Semester	**3**	**0**	**0**	**3**

Prerequisites: Familiarity with Computer Networks and Information Security.

Course Objectives: The course will focus on the models, tools, and the techniques for enforcement of cyber security policies, with some emphasis on the use of cryptography.

Syllabus:

UNIT I: (8 Lectures)

Introduction to Computer Security: Definition, Threats to security, Government Requirements, Information Protection and Access Controls, Computer Security Efforts, Standards, Computer Security Mandates and Legislation, Privacy Considerations, International Security Activity.

UNIT II: (7 Lectures)

Cyber Crime Issues: Unauthorized Access to Computers, Computer Intrusions, White Collar Crimes, Viruses and Malicious Code, Internet Hacking and Cracking, Virus Attacks, Pornography, Software Piracy, Intellectual Property, Mail Bombs, Exploitation, Stalking and Obscenity in Internet, Digital Laws and Legislation, Law Enforcement Roles and Responses.

UNIT III: (10 Lectures)

Secure System Planning and Administration: Introduction to the Orange Book, Security Policy Requirements, Accountability, Assurance and Documentation Requirements, Network Security, The Red Book and Government Network Evaluations.

Information Security Policies and Procedures: Corporate Policies, Tier 1, Tier 2 and Tier3 Policies, Process Management, Planning and Preparation, Developing Policies, Asset Classification Policy, Developing Standards.

UNIT IV: (8 Lectures)

Information Security: Fundamentals, Employee Responsibilities, Information Classification, Information Handling, Tools of Information Security, Information Processing, Secure Program Administration.

UNIT V: (7 Lectures)

Organizational and Human Security: Adoption of Information Security Management Standards, Human Factors in Security, Role of Information Security Professionals.

Course Outcomes:

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

CO 1. Specify tools and architectures to help secure computer and information systems both proactively and reactively.

CO 2. Describe how cyber attacks against an organization can be monitored and investigated for actionable intelligence.

CO 3. Apply skills and knowledge to create new responses to emerging cyber security problems so that they can respond to new attacks as they evolve.

CO 4. Identify components of a modern information system and the threats that challenge their security.

CO 5. Identify the risks an organization faces due to cyber threats and recommend steps to combat those risks.

Mapping of COs to POs:

POs/ COs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	3	2	-	-	-	-	-	2	-	-	-
CO2	3	2	-	2	2	-	-	3	-	-	-
CO3	3	3	3	3	2	-	2	-	3	-	3
CO4	3	2	-	-	-	-	-	-	-	-	-
CO5	3	2	-	3	2	-	-	-	-	-	-

TEXT BOOKS:

1. Debby Russell and Sr. G.T Gangemi, "Computer Security Basics (Paperback)", 2nd Edition, O' Reilly Media.
2. Nelson Phillips and Enfinger Steuart, "Computer Forensics and Investigations", Cengage Learning, New Delhi.
3. Thomas R. Peltier, "Information Security policies and procedures: A Practitioner's Reference", 2nd Edition, Prentice Hall.

REFERENCE BOOKS:

1. Kenneth J. Knapp, "Cyber Security and Global Information Assurance: Threat Analysis and Response Solutions", IGI Global.
2. Thomas R Peltier, Justin Peltier and John blackley, "Information Security Fundamentals", 2nd Edition, Prentice Hall.
3. Kevin Mandia, Chris Prosise, Matt Pepe, "Incident Response and Computer Forensics ", Tata McGraw -Hill, New Delhi.

GO PROGRAMMING
(ELECTIVE–II)

Subject Code: PGCS2T1722	**L**	**T**	**P**	**C**
I Year / II Semester	**3**	**0**	**0**	**3**

Prerequisites: Familiarity with any programming language.

Course Objectives:
The course is designed to quickly cover the basics, and then dive into more advanced features of the Go programming language.

Syllabus:

UNIT I: (7 Lectures)

Introduction: Origins and evolution, Languages that influenced Go, Why a new language?, Targets of the language, Guiding design principles, Characteristics of the language, Uses of the language, Missing features, Programming in Go.

Program Structure: Names, Declarations, Variables, Assignments, Type Declarations, Packages and Files, Scope.

UNIT II: (10 Lectures)

Basic Data Types: Integers, Floating-Point Numbers, Complex Numbers, Booleans, Strings, Constants. **Control Structures:** if else construct, switch construct, for construct, break, continue and labels.

Composite Types: Arrays, Slices, Maps, Structs, JSON,Text and HTML Templates.

Functions: Function Declarations, Recursion, Multiple Return Values, Errors, Function Values, Anonymous Functions, Variadic Functions, Deferred Function Calls, Panic, Recover.

UNIT III: (10 Lectures)

Methods: Method Declarations, Methods with a Pointer Receiver, Composing Types by Struct Embedding, Method Values and Expressions, Encapsulation.

Interfaces: Interfaces as Contracts, Interface Types, Interface Satisfaction, Parsing Flags with flag.Value, Interface Values, The error Interface, Type Assertions, Discriminating Errors with Type Assertions, Querying Behaviors with Interface Type Assertions, Type Switches.

UNIT IV: (10 Lectures)

Reading and Writing: Reading input from the user, Reading from and writing to a file, Copying files, Reading arguments from the command-line, Reading files with a buffer, Reading and writing files with slices, Using defer to close a file.

Goroutines and Channels: Goroutines, Concurrent Clock Server, Concurrent Echo Server, Channels, Looping in Parallel, Concurrent Web Crawler, Multiplexing with select, Concurrent Directory Traversal, Cancellation, Chat Server.
Concurrency with Shared Variables: Race Conditions, Mutual Exclusion, Read/Write Mutexes, Memory Synchronization, Lazy Initialization, The Race Detector, Concurrent Non Blocking Cache, Goroutines and Threads.

UNIT V: (8 Lectures)

Packages and Go Tool: Introduction, Import Paths, The Package Declaration, Import Declarations, Blank Imports, Packages and Naming, The Go Tool.
Testing: Go Test Tool, Test Functions, Coverage, Benchmark Functions, Profiling, Example Functions.

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

- **CO 1.** Study the basic constructs of Go Programming and learn its structural elements in detail.
- **CO 2.** Develop modular programming and make use of functions and methods.
- **CO 3.** Implement the Interfaces and Goroutines for executing the program independently and simultaneously.
- **CO 4.** Apply Go's concurrency model to build massively parallel systems and examine the different packages in Go.

Mapping of COs to POs:

POs/ COs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	3	-	-	-	-	-	-	-	-	-	-
CO2	3	3	3	-	3	3	3	3	-	-	3
CO3	3	3	3	-	3	3	3	3	-	-	3
CO4	3	3	3	-	-	-	-	-	-	-	-

TEXT BOOKS:

1. Alan A. A. Donovan, Brian W. Kernighan, "The Go Programming Language", Addison-Wesley.
2. Ivo Balbaert, "The Way to GO – A Thorough Introduction to the Go Programming Language", i-Universe Publisher.

REFERENCE BOOKS:

1. Mark Summerfield, Programming in Go: Creating applications for the 21st century. Addison-Wesley.
2. Caleb Doxsey, An Introduction to Programming in Go.
3. Tarik Guney, "Hands-On Go Programming: Explore Go by solving real-world challenges", Packt Publishing.
4. John P. Baugh, "Go Programming", CreateSpace Publisher.
5. Mat Ryer, Go Programming Blueprints, Packt Publishing.

DEEP LEARNING
(ELECTIVE–III)

Subject Code: PGCS2T0822	**L**	**T**	**P**	**C**
I Year / II Semester	**3**	**0**	**0**	**3**

Prerequisites: Familiarity with Probability & Statistics, Design and Analysis of Algorithms.

Course Objectives: The objective of the course is to provide exposure to these advances and facilitate in depth discussions on deep learning.

Syllabus:

UNIT I: (8 Lectures)

Machine Learning Basics

Learning Algorithms, Capacity, Over fitting and Under fitting, Hyper parameters and Validation Sets, Estimators, Bias and Variance, Maximum Likelihood, Estimation Bayesian Statistics.

Supervised Learning Algorithms, Unsupervised Learning Algorithms, Stochastic Gradient Descent, Building a Machine Learning Algorithm, Challenges Motivating Deep Learning.

UNIT II: (8 Lectures)

Deep Feedforward Networks

Example: Learning XOR, Gradient-Based Learning, Hidden Units, Architecture Design, Back-Propagation and Other Differentiation Algorithms.

UNIT III: (10 Lectures)

Regularization for Deep Learning

Parameter Norm Penalties, Norm Penalties as Constrained Optimization, Regularization and Under-Constrained Problems, Dataset Augmentation, Noise Robustness, Semi-Supervised Learning, Multitask Learning.

Optimization for Training Deep Models

How Learning Differs from Pure Optimization, Challenges in Neural Network Optimization, Basic Algorithms, Parameter Initialization Strategies, Algorithms with Adaptive Learning Rates, Approximate Second-Order Methods, Optimization Strategies and Meta-Algorithms.

UNIT IV: (10 Lectures)

Convolutional Networks

The Convolution Operation, Motivation, Pooling, Convolution and Pooling as an Infinitely Strong Prior, Efficient Convolution Algorithms, Random or Unsupervised Features, The Neuroscientific Basis for Convolutional Networks.

UNIT V: (8 Lectures)

Sequence Modeling: Recurrent and Recursive Nets

Unfolding Computational Graphs, Recurrent Neural Networks, Bidirectional RNNs, Deep Recurrent Networks, Recursive Neural Networks, The Challenge of Long-Term Dependencies, Optimization for Long-Term Dependencies, Explicit Memory.

Course Outcomes:

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

CO 1. Demonstrate the basics of Machine Learning.

CO 2. Analyze the importance of deep feed forward networks.

CO 3. Summarize the significance of regularization for Deep Learning.

CO 4. Implement optimization in DL.

CO 5. Perceive the importance of Convolutional Networks and its significance.

CO 6. Illustrate the knowledge on Sequence Modeling.

Mapping of COs to POs:

POs/ COs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	3	3	-	-	-	-	-	3	-	-	-
CO2	3	3	-	2	-	-	-	-	-	-	-
CO3	3	3	-	-	-	-	-	-	-	-	-
CO4	3	3	3	3	3	3	-	-	-	3	3
CO5	3	3	-	3	-	-	-	-	-	-	-
CO6	3	3	-	-	-	-	-	-	-	-	-

TEXT BOOKS:

1. Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. Deep learning, Vol. 1. Cambridge: MIT press.
2. François Duval , Deep Learning: Deep Learning for Beginners. Practical Guide with Python and Tensorflow, Data Sciences Publishing.

REFERENCE BOOKS:

1. Sebastian Raschka, Vahid Mirjalili, Python Machine Learning: Machine. Learning and Deep Learning with Python, scikit-learn, and TensorFlow, 2nd Edition, Packt Publishing.

BIG DATA TECHNOLOGIES
(ELECTIVE-III)

Subject Code: PGCS2T0922	**L**	**T**	**P**	**C**
I Year / II Semester	**3**	**0**	**0**	**3**

Prerequisites: The student should have knowledge of high level programming languages and SQL for analyzing the data.

Course Objectives: The student will be able to understand Big Data as a popular term used to describe the exponential growth, availability and use of information, both structured and unstructured. It is imperative that organizations and IT leaders focus on the ever-increasing volume, variety and velocity of information that forms Big Data. Hadoop is the core platform for structuring BigData, and solves the problem of making it useful for Analytics.

Syllabus:

UNIT I: (8 Lectures)

Introduction to Big Data: What is Big Data and where it is produced? Rise of Big Data, Compare Hadoop vs traditional systems, Limitations and Solutions of existing Data Analytics Architecture, Attributes of Big Data, Types of Data, Use Cases of Big Data, Other technologies vs Big Data.

UNIT II: (9 Lectures)

Hadoop Architecture and HDFS: What is Hadoop? Hadoop History, Distributing Processing System, Core Components of Hadoop, HDFS Architecture, Hadoop Master – Slave Architecture, Daemon Types, Name node, Data node, Secondary Name node.

Hadoop Clusters and the Hadoop Ecosystem- What is Hadoop Cluster? Pseudo Distributed mode, Type of Clusters, Hadoop Ecosystem: Pig, Hive, Flume, SQOOP.

UNIT III: (9 Lectures)

Hadoop MapReduce Framework: Overview of MapReduce Framework, MapReduce Architecture, Job Tracker and Task Tracker, Use Cases of Map Reduce, Anatomy of Map Reduce Program.

MapReduce Programs in Java: Basic MapReduce API Concepts, Writing MapReduce Driver, Mappers, and Reducers in Java, Speeding up Hadoop Development by Using Eclipse, Word Count Example and Weather Dataset Example.

UNIT IV: (10 Lectures)

Hive and HiveQL- What is Hive? Hive vs MapReduce, Hive DDL : Create/Show/Drop Tables, Internal and External Tables, Hive DML : Load Files & Insert Data, Hive Architecture & Components, Difference between Hive and RDBMS, Partitions in Hive.

Pig: Pig vs MapReduce, Pig Architecture & Data types, Shell and Utility components, Pig Latin Relational Operators, Pig Latin: File Loaders and UDF, Programming structure in UDF, Pig Jars Import and limitations of Pig.

UNIT V: (9 Lectures)

Apache SQOOP: Why and What is SQOOP?, SQOOP Architecture, Benefits of SQOOP, Importing Data Using SQOOP.

Apache Flume: Introduction, Flume Model and Goals, Features of Flume, Flume Use Cases.

Course Outcomes:

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

CO 1. Outline importance of Big Data in solving real time problems in data analytics.

CO 2. Illustrate Hadoop ecosystem and its components in detail.

CO 3. Make use of distributed file systems and Hadoop and can write MapReduce programs to solve complex problems.

CO 4. Explore the Hadoop ecosystems core components and apply in real-time scenarios.

Mapping of COs to POs:

POs/ COs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	3	3	3	-	-	-	-	-	-	-	-
CO2	3	3	-	-	-	-	-	-	-	-	-
CO3	3	3	3	-	3	3	3	-	-	-	3
CO4	3	3	3	3	3	-	-	-	3	-	-

TEXT BOOKS:

1. Tom White, Hadoop : The Definitive Guide, 3rd Edition, O'reilly

2. Dirk deRoos, Chris Eaton, George Lapis, Paul Zikopoulos, Tom Deutsch, "Understanding Big Data Analytics for Enterprise Class Hadoop and Streaming Data", 1st Edition, TMH.

REFERENCE BOOKS:

1. Alex Holmes, Hadoop in Practice, MANNING Publications.
2. Srinath Perera, Thilina Gunarathne, Hadoop MapReduce Cookbook, Packt publishing.

ROBOTIC PROCESS AUTOMATION
(ELECTIVE–III)

Subject Code: PGCS2T1822	**L**	**T**	**P**	**C**
I Year / II Semester	**3**	**0**	**0**	**3**

Prerequisites: A little bit of C programming knowledge, analytical and logical thought procedure to build a process is required.

Course Objectives: This course will give you an overview of robotic process automation (RPA) technology. You will learn the characteristics, benefits, risks, and challenges of RPA. You will learn about the RPA landscape, how RPA is transforming businesses and how it is affecting accounting and finance professionals.

Syllabus:

UNIT I: (8 Lectures)
RPA Foundations: RPA, Flavors of RPA, History of RPA, Benefits of RPA, Downsides of RPA, RPA Compared to BPO, BPM, and BPA; Consumer Willingness for Automation, Workforce of the Future and RPA Skills.
Planning: RPA Consulting: Some Case Studies, What to Automate? ROI for RPA, RPA Use Cases, The Plan and RPA Vendor Evaluation.

UNIT II: (9 Lectures)
Center of Excellence: CoE, Need of CoE, Forming the Team, Business Analyst, Developer, RPA Solution Architect, RPA Supervisor, What Should a CoE Do?, Communication, Change Management.
Bot Development: Installation of UiPath, Activities, Flowcharts and Sequences, Log Message, Variables, Loops and Conditionals, Switch, Debug, Common UiPath Functions, The UiPath Orchestrator, Best Practices.

UNIT III: (10 Lectures)
Deployment and Monitoring : Testing, Going into Production, Monitoring, Security, Scaling. **Data Preparation:** Types of Data, Big Data, Issues with Big Data, Data Process, Types of Algorithms, Bias and Open Source RPA.
Using Blue Prism: Building the first Blue Prism process, Pages, Data Items, Blocks, Collections, Loops, Actions, Decisions, Choices and Calculations.

UNIT IV: (9 Lectures)
Implementing Business Objects: Creating a business object, Business Studio, Renaming actions, Application Modeller, Using the Navigate stage, Publishing an action, Using a custom Business Object from a process.
Spying Elements: Spying elements on a web page, How does spying work?, Tweaking and Tightening the match criteria, Adding and Categorizing elements, More spy modes, UI Automation mode, UI Automation navigator, Surface automation with region mode.

Write, Wait, and Read: Creating the search action, Writing to text boxes, Clicking buttons, Wait stage, Read stage, Reading the search results, Using dynamic match attributes.

UNIT V: (9 Lectures)

Excel & Email Automation: Reading the shopping list, Importing the Excel VBO, Using MS Excel VBO, Opening an Excel file, Reading an entire worksheet into a collection, Writing to a cell, Considerations for CSV, Sending and Receiving Emails.

Control Room and Work Queue: Publishing a process, Running a process, Scheduling processes and work queues.

Exception Handling: Expected and unexpected exceptions, Raising exceptions, Handling exceptions, Debugging and troubleshooting from the control room.

Course Outcomes:

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

CO1. Understand the different RPA tools and its architecture for process development.

CO2. Acquire the basic knowledge on UiPath and Blue Prism softwares.

CO3. Apply the different stages to create and demonstrate static processes.

CO4. Demonstrate the Blue Prism business studio and its stages by creating real time applications.

CO5. Classify the exception handling and error management techniques with different stages in RPA.

Mapping of COs to POs:

POs/ COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	-	-	-	-	-	-	-	-	-	-
CO2	3	-	-	-	-	-	-	-	-	-	-
CO3	2	2	3	-	3	-	-	-	-	-	-
CO4	2	3	-	-	3	-	-	3	-	-	-
CO5	3	3	-	-	-	-	-	-	-	-	-

TEXT BOOKS:

1. Tom Taulli, The Robotic Process Automation Handbook: A Guide to Implementing RPA Systems, Apress.
2. Lim Mei Ying, Robotic Process Automation with Blue Prism Quick Start Guide, Packt Publishing.

REFERENCE BOOKS:

1. Alok Mani Tripathi, Learning Robotic Process Automation, Packt Publishing Ltd.
2. Kelly Wibbenmeyer, The Simple Implementation Guide to Robotic Process Automation (RPA): How to Best Implement RPA in an Organization, iUniverse.

AUGMENTED REALITY AND VIRTUAL REALITY
(ELECTIVE–III)

Subject Code: PGCS2T1922	**L**	**T**	**P**	**C**
I Year / II Semester	**3**	**0**	**0**	**3**

Prerequisites: The students should have basic knowledge on programming and computer graphics.

Course Objectives:
To introduce the basic concepts of Augmented Reality and Virtual Reality and to gain knowledge on various devices required for interaction and applications.

Syllabus:

UNIT I: (8 Lectures)
Introduction: Virtual Reality, Augmented Reality, Mixed Reality, Augmented Virtuality, Extended Reality, History, VR Features, VR Controllers, Current issues with VR, AR Mobile devices, AR headsets, AR glasses, AR Controllers, Current issues with AR.

UNIT II: (8 Lectures)
Consuming Content in VR : High-end devices, Mid-tier devices, Low-end devices, Near-Future Hardware.
Consuming Content in AR: Microsoft HoloLens, Meta 2, Magic Leap, Mira Prism, Apple ARKit, Google ARCore, Near-Future Hardware.

UNIT III: (10 Lectures)
Creating Content in VR and AR: Evaluating Your Project, Planning Your Virtual Reality Project, Planning Your Augmented Reality Project, Assessing Design Software, Capturing Real Life, Assessing Development Software, Distributing Your Content.
Cross-Platform Theory: Role of Game Engines, Understanding 3D Graphics, The Virtual Camera, Degrees of Freedom, Portability Lessons from Video Game Design, Simplifying the Controller Input.

UNIT IV: (9 Lectures)
Virtual Reality Toolkit: History of VRTK, SteamVR Unity Toolkit, VRTK v4, Future of VRTK, Success of VRTK, Getting Started with VRTK 4.
Best Practices: Handling Locomotion in VR & AR, Effective Use of Audio in VR & AR, Common Interactions Paradigms, Inventory for VR, Augmented Reality Raycasts.

UNIT V: (8 Lectures)
Applications: Travel, Museums, Aerospace, Retail, Military, Education, Entertainment, Real Estate, Advertising and Marketing, Mobile Apps for Experiencing Augmented Reality, Future of Virtual Reality and Augmented Reality.

Course Outcomes:

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

CO 1. Gain knowledge on AR & VR and various components involved in manifesting the same.

CO 2. Plan content creation and identify necessary software required in implementing AR & VR.

CO 3. Analyze the portability issues and understand the best practices.

CO 4. Understand how to implement various applications using AR and VR technologies.

Mapping of COs to POs:

POs/ COs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	3	-	-	-	-	-	-	-	-	-	-
CO2	3	-	-	-	-	-	-	-	-	-	-
CO3	3	3	-	-	-	-	-	-	3	-	-
CO4	3	-	-	-	-	-	-	-	-	-	-

TEXT BOOKS:

1. Paul Mealy, Virtual & Augmented Reality For Dummies, John Wiley & Sons, Inc

2. Erin Pangilinan, Steve Lukas and Vasanth Mohan, Creating Augmented and Virtual Realities, O'Reilly Media Inc.

REFERENCE BOOKS:

1. Kelly S. Hale, Kay M. Stanney, Handbook of Virtual Environments: Design, Implementation, and Applications, Second Edition, CRC Press.

2. Gregory C. Burdea & Philippe Coiffet, John, Virtual Reality Technology, Second Edition, Wiley & Sons, Inc.

3. William R.Sherman, AlanCraig, Understanding Virtual Reality, interface, Application and Design, Elsevier (Morgan Kaufmann).

4. John Vince, Virtual Reality Systems, Pearson Education.

5. Andrew Davison, Killer Game Programming in Java, Oreilly-SPD.

6. Alan B Craig, William R Sherman and Jeffrey D Will, "Developing Virtual Reality Applications: Foundations of Effective Design", Morgan Kaufmann.

7. Alan B. Craig, Understanding Augmented Reality, Concepts and Applications, Morgan Kaufmann

8. Steve Aukstakalnis, "Practical Augmented Reality: A Guide to the Technologies, Applications, and Human Factors for AR and VR", Addison Wesley.

9. Brett S. Martin, "Virtual Reality", Norwood House Press.

10. Anand R., "Augmented and Virtual Reality", Khanna Publishing House, Delhi

11. Adams, "Visualizations of Virtual Reality", Tata McGraw Hill.

FULL STACK DEVELOPMENT LAB

Subject Code: PGCS2P1322	**L**	**T**	**P**	**C**
I Year / II Semester	**0**	**0**	**3**	**1.5**

Prerequisites: Students should have the prior knowledge on HTML, CSS and JavaScript technologies.

Course Objectives: This lab course is aimed to get the hands on experience on complete Full Stack development(MERN) by imparting core concepts of both the frontend and backend development with latest web technologies and NOSQL Databases.

List of Experiments:

1. Develop a React render HTML Student Registration Webpage.
2. Incorporate styles to Student Registration Webpage using React CSS.
3. Create a reusable React Component and implement in any webpage.
4. Develop a React Form that performs the actions click, change, mouseover using React Events.
5. Install Node.js, create Node.js application, make a request to the Node.js Server and display "Hello World" in web browser.
6. Create own modules in Node.js application to process images and to work with Dates, Strings and Colors.
7. Create Node.js application to perform Create, Read, Update and Delete operations on Files.
8. Design a Node.js application that refreshes automatically after using Express.js
9. Perform Create, Read, Update and Delete (CRUD) Operations in MongoDB NOSQL Platform.
10. Develop a Node.js application to store and retrieve the details of students using MongoDB.
11. Design the front-end for a Shopping webpage using AngularJS.

Course Outcomes:

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

- **CO1.** Set-up React and AngularJS environments and design a frontend for Web Pages.
- **CO2.** Create Node.js framework and develop interactive Web applications using the skills of Node.js and Express.js
- **CO3.** Install MongoDB and integrate with the web applications to implement real world applications.

Mapping of COs to POs:

POs/ COs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	3	3	3	3	3	-	-	-	-	-	-
CO2	3	3	3	3	3	-	-	-	-	-	-
CO3	3	3	3	3	3	-	-	-	-	-	-

TEXT BOOKS:

1. Azat Mardan, React Quickly, Manning Publications Co.

2. Colin J. Ihrig and Adam Bretz, Full Stack JavaScript Development with MEAN, SitePoint Pty Ltd.

REFERENCE BOOKS:

1. Kirupa Chinnathambi, Learning React, Pearson Education Inc.

2. Cássio de Sousa Antonio, Pro React, Apress.

3. Vasan Subramanian, Pro MERN Stack, Apress Publications.

4. Simon Holmes, Clive Harber, Getting Mean with Mongo, Express, Angular and Node, Manning Publications Co.

5. Amos Q. Haviv, MEAN Web Development, Packt Publishing.

SOFTWARE TESTING AND DESIGN PATTERNS LAB

Subject Code: PGCS2P2022	**L**	**T**	**P**	**C**
I Year / II Semester	**0**	**0**	**3**	**1.5**

Prerequisites: Familiarity with Software Engineering.

Course Objectives: This course is designed to enable a clear understanding of the foundations, techniques, and tools in the area of software testing and its practice.

SOFTWARE TESTING LAB:

1. Study of various tools such as
    a. Study of any testing tool (e.g. Win runner)
    b. Study of any web testing tool (e.g. Selenium)
    c. Study of any bug tracking tool (e.g. Bugzilla, bugbit)
    d. Study of any test management tool (e.g. Test Director)
    e. Study of any open source-testing tool (e.g. Test Link)
2. Perform experiments to do the following:
    a. Requirements Testing
    b. Use – case Scenario Testing
    c. Unit Testing
    d. Regression Testing
    e. Integration Testing
    f. Validation Testing
    g. Acceptance Testing
    h. System Testing
3. Prepare test plan and develop test case hierarchy.
4. Generate Test cases and Test Documentation in the following case studies
    a. Library System
    b. Course Registration System
    c. Implement a Quiz System
    d. Student Marks Analyzing System
    e. Online Ticket Reservation System

DESIGN PATTERNS LAB:

1. Using UML design Abstract factory design pattern
2. Using UML design Builder Design pattern
3. Using UML design Facade Design pattern
4. Using UML design Bridge Design pattern
5. Using UML design Decorator Design pattern
6. User gives a print command from a word document. Design to represent this chain of responsibility design pattern.

Course Outcomes:

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

CO1: Understand fundamental concepts in software testing, including software testing process, criteria, strategies, tools and methods.

CO2: Execute specific software tests with well-defined objectives and targets.
CO3: Design and implement comprehensive test plans and implement test cases for case studies.
CO4: Design and develop solutions using the design patterns.

Mapping of COs to POs:

POs/ COs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	2	-	-	-	-	3	-	3	-	-	-
CO2	3	3	-	-	3	3	-	3	-	-	3
CO3	3	-	-	-	3	3	-	3	-	-	3
CO4	3	-	-	-	3	3	-	3	3	-	3

TEXT BOOKS:

1. P. Nageswara Rao, Software Testing Concepts and Tools, Dream Tech Press.
2. Nageshwara Rao Pusuluri, Software Testing Concepts and Tools by Dream Tech Press.
3. Erich Gamma, Design Patterns, Pearson Education.

REFERENCE BOOKS:

1. K. V. K. K. Prasad, Software Testing Tools, Dream Tech Press.
2. S. Subashini, N. Satheesh kumar, Software Testing with Visual Studio Team System, Shroff Publishers Distributors.