

CURRICULUM
Academic year 2026 – 2027
Batch (2026-2028)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

I - IV Semester M. Tech

COMPUTER SCIENCE AND ENGINEERING

Deparment of Computer Science and Engineering

RAMAIAH INSTITUTE OF TECHNOLOGY
(Autonomous Institute, Affiliated to VTU)
BANGALORE – 560054

About the Institute:

Ramaiah Institute of Technology (RIT), Bengaluru, established in **1962** under the **Gokula Education Foundation (GEF)** by the visionary philanthropist **Late Dr. M. S. Ramaiah**, is one of India's premier autonomous engineering institutions. Founded with the vision of providing quality technical education that addresses global and societal needs, the institute has grown into a centre of excellence in engineering, architecture, management, science, research, and innovation.

The institute offers **17 undergraduate**, **15 postgraduate**, and **doctoral programmes** across diverse disciplines. All programmes are approved by the **All India Council for Technical Education (AICTE)**, and all eligible academic departments are accredited by the **National Board of Accreditation (NBA)**. The institute is accredited with an **'A+' Grade by the National Assessment and Accreditation Council (NAAC)**. RIT has enjoyed **autonomous status** for its undergraduate and postgraduate programmes under **Visvesvaraya Technological University (VTU)** since 2007 and has also been granted autonomy for its full-time **Ph.D. programmes**.

RIT is recognised for its strong research ecosystem, supported by **more than 380 faculty members**, nearly **65–67% of whom hold doctoral degrees**, and **over 300 Ph.D. scholars**. The institute promotes interdisciplinary research through sponsored projects and Centres of Excellence, including the 5G use cases lab, **Centre for Imaging Technologies**, **Centre for Advanced Materials Technology**, **Centre for Antennas and Radio Frequency Systems (CARFS)**, **Centre for Cyber Physical Systems**, **Schneider Centre of Excellence**, and the **Centre for Bio and Energy Materials Innovation**. The institute also records an average of **approximately 350 research publications annually**.

The campus provides world-class infrastructure comprising state-of-the-art laboratories, modern classrooms, advanced computing facilities, a central library, and a digital library with access to major scholarly resources, including **IEEE Xplore, Elsevier ScienceDirect, SpringerLink, and Taylor & Francis**. Students benefit from excellent sports facilities, modern hostels, auditoriums, technology-enabled conference halls, and a comprehensive mentoring and placement ecosystem.

Innovation and entrepreneurship are fostered through the **Entrepreneurship Development Cell (EDC)** and **Ramaiah Evolute**, the institute's Section 8 company supporting startup incubation and technology commercialization. RIT has established collaborations with **46 industries**, enabling consultancy, internships, collaborative research, and skill development.

The institute has consistently earned national recognition for academic excellence. In the **NIRF 2025** rankings, RIT secured **75th rank in Engineering**, and **31st rank in Architecture**. With its strong academic foundation, vibrant research culture, industry partnerships, excellent placement record, and distinguished alumni network, Ramaiah Institute of Technology continues to be a preferred destination for quality technical education and innovation in India.

About the Department:

Year of Establishment	1984
Names of the Programmes offered	1. UG: B.E. in Computer Science and Engineering 2. PG: M.Tech. in Computer Science and Engineering 3. PG: M.Tech. in Computer Network Engineering 4. Ph.D. (under VTU research Center) 5. Ph.D. (Ramaiah Doctoral Fellowship) 6. M.Sc(Engg.) by Research (under VTU research Center)

The Department of CSE at Ramaiah Institute of Technology (RIT), Bengaluru, is committed to excellence in education, research, innovation, and industry collaboration. The department is supported by an eminent emeritus professor, 19 faculty members with Ph.D. degrees, and 12 faculty members pursuing doctoral research, who actively contribute to teaching, research, and institutional development.

Recognized as a Research Centre under VTU, the department currently has 83 Ph.D. scholars, with 53 doctoral degrees awarded. During the last three years, the department has published 114 international journal papers, 100 international conference papers, and 36 book chapters. It has also filed 30 patents, with 3 patents granted, reflecting its strong focus on innovation and intellectual property.

The department supports research through five externally funded projects, several faculty-mentored student innovation projects, and a departmental library with over 800 technical books. Faculty and students have received 21 and 66 awards, respectively, during the past three years. Strong industry collaborations with organizations such as SAP, IBM, and HP facilitate consultancy, collaborative research, internships, and skill development.

The department offers industry-oriented learning through state-of-the-art laboratories and Centres of Excellence, including the SAP Laboratory, IBM Centre of Excellence, and CUDA Learning Centre. The undergraduate program is NBA accredited for six years (2022), while the M.Tech. CSE program is accredited for 2024–2027.

To enhance technical and professional competencies, the department regularly organizes seminars, workshops, hackathons, expert lectures, and industry interactions. Students are encouraged to pursue MOOCs through NPTEL, Coursera, and TedX, with over 850 certifications earned. The department also promotes holistic development through six professional student clubs, extracurricular and sports activities, and a vibrant alumni network supporting mentoring, internships, placements, and collaborative initiatives.

VISION OF THE INSTITUTE

To be an Institution of International Eminence, renowned for imparting quality technical education, cutting edge research and innovation to meet global socio- economic needs

MISSION OF THE INSTITUTE

M S Ramaiah Institute of Technology shall meet the global socio-economic needs through

1. Imparting quality technical education by nurturing a conducive learning environment through continuous improvement and customization
2. Establishing research clusters in emerging areas in collaboration with globally reputed organizations
3. Establishing innovative skills development, techno-entrepreneurial activities and consultancy for socio-economic needs

QUALITY POLICY

We at M. S. Ramaiah Institute of Technology, Bangalore strive to deliver comprehensive, continually enhanced, global quality technical and management education through an established Quality Management System complemented by the synergistic interaction of the stake holders concerned.

VISION OF THE DEPARTMENT

To build a strong learning and research environment in the field of Computer Science and Engineering that promotes innovation towards betterment of the society.

MISSION OF THE DEPARTMENT

1. To produce Computer Science post graduates who, trained in design and implementation of computational systems through competitive curriculum and research in collaboration with industry and research organizations.
2. To educate students in technology competencies by providing professionally committed faculty and staff.
3. To inculcate strong ethical values, leadership abilities and research capabilities in the minds of students so as to work towards the progress of the society.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs):

PEO1 Pursue a successful career in the field of Computer Science & Engineering or a related field utilizing his/her education and contribute to theprofession as an excellent employee, or as an entrepreneur.

PEO2 Be aware of the developments in the field of Computer Science & Engineering, continuously enhance their knowledge informally or by pursuing doctoral studies and engage in research and inquiry leading to new innovations and products.

PEO3 Be able to work effectively in multidisciplinary and multicultural environments and be responsible members and leaders of their communities.

PEO4 Understand the human, social and environmental context of their profession and contribute positively to the needs of individuals and society at large.

PROGRAM OUTCOMES (POs):

PO1 An ability to independently carry out research / investigation and developmentwork to solve practical problems.

PO2 An ability to write and present a substantial technical report / document.

PO3 Students should be able to demonstrate a degree of mastery over Computer Science and Engineering. The mastery should be at a level higher than the requirements in the appropriate bachelor program.

PO4 Acquire professional and intellectual integrity to stress upon the impact of computer engineering applications with respect to economic and environmental aspects.

PO5 Acquire methods of engaging in life-long learning not only to predict and plan the projects of the future but also to groom others in the group.

PG Curriculum Course Credits
Scheme Structure of M. Tech. Program from the Academic year 2025-26 (Batch 2025-27)

Semester	Professional Core Courses (PCC)	Professional Core Courses Lab (PCCL)	Professional Courses - Electives (PEC)	Mandatory Credit Course (MCC)	Project Work / Seminar (PW)	Internship (INT)	Total semester load
First	11	2	6	3	-	-	22
Second	12	2	9		-	-	23
Third	-	-	6		6	5	17
Fourth	-	-	-		18	-	18
Total	**23**	**4**	**21**	**3**	**24**	**5**	**80**

M.Tech. in Computer Science and Engineering Scheme of Teaching and Examination Batch 2026-28									
I SEMESTER									
Sl. No.	**Subject Code**	**Subject**	**Teaching Department**	**Category**	**Credits**				**Total contact hours / week**
					L	**T**	**P**	**Total**	
1	MCS11	Advanced Engineering Mathematics	Mathematics	Math	2	1	0	3	4
2	MCS12	Advanced Database Management Systems	CSE	PCC	4	0	0	4	4
3	MCS13	Advanced Computational Intelligence	CSE	PCC	4	0	0	4	4
4	MCSE14x	Professional Elective Course - I	CSE	PEC	3	0	0	3	3
5	MCSE15x	Professional Elective Course - II	CSE	PEC	3	0	0	3	3
6	RMI16	Research Methodology and IPR*	CSE	MCC	3	0	0	3	3
7	MCS L17	Advanced Database Laboratory	CSE	PCCL	0	0	1	1	2
8	MCS L18	Application development using Python Programming Laboratory	CSE	PCCL	0	0	1	1	2
Total					**19**	**1**	**2**	**22**	**25**

Where x=1,2,3…L –Lecture, T – Tutorial, P- Practical. Note: * Common curriculum across all branches

Professional Elective Course I		**Professional Elective Course II**	
Subject Code	**Subject Name**	**Subject Code**	**Subject Name**
MCSE141	Cryptography and Network Security	MCSE151	Data Science and Management
MCSE142	Advanced Distributed Systems	MCSE152	IoT and Edge AI
MCSE143	Extended Reality Systems	MCSE153	Image Processing and Machine Vision

M.Tech. in Computer Science and Engineering Scheme of Teaching and Examination Batch 2026-28									
II SEMESTER									
Sl. No.	**Subject Code**	**Subject**	**Teaching Department**	**Category**	**Credits**				**Total contact hours /week**
					L	**T**	**P**	**Total**	
1	MCS21	Advanced Data Structures and Algorithms	CSE	PCC	3	1	0	4	5
2	MCS22	Advanced Operating Systems	CSE	PCC	4	0	0	4	4
3	MCS23	Cloud Computing and Micro services	CSE	PCC	4	0	0	4	4
4	MCSE24x	Professional Elective Course – III	CSE	PEC	3	0	0	3	3
5	MCSE25x	Professional Elective Course – IV	CSE	PEC	3	0	0	3	3
6	MCSE26x	Professional Elective Course – V	CSE	PEC	3	0	0	3	3
7	MCSL27	Cloud Computing and Micro services Laboratory	CSE	PCCL	0	0	1	1	2
8	MCSL28	System Programming Laboratory	CSE	PCCL	0	0	1	1	2
Total					**20**	**1**	**2**	**23**	**26**

Where x=1,2,3…L –Lecture, T – Tutorial, P- Practical

Professional Elective Course III		**Professional Elective Course IV**		**Professional Elective Course V**	
Subject Code	**Subject Name**	**Subject Code**	**Subject Name**	**Subject Code**	**Subject Name**
MCSE241	Advanced Software Engineering and DevOps	MCSE251	Wireless and Mobile Networks	MCSE261	Natural Language Processing
MCSE242	Cyber-Physical Systems and Smart Networks	MCSE252	Game Theory and Mechanism Design for Computing Systems	MCSE262	Full Stack Development
MCSE243	Blockchain Essentials and Dapps	MCSE253	Quantum Computing for Computer Scientists	MCSE263	Information Retrieval and Search Systems

M.Tech. in Computer Science and Engineering
Scheme of Teaching and Examination Batch 2026-28

III SEMESTER

Sl. No.	Subject Code	Subject	Teaching Department	Category	Credits				Total contact hours /week
					L	T	P	Total	
1	MCSE31x	Professional Elective Course – VI*	CSE	PEC	3	0	0	3	3
2	MCSE32x	Professional Elective Course – VII*	CSE	PEC	3	0	0	3	3
3	MCSI33	Internship/Industrial Training	CSE	INT	0	0	5	5	5
4	MCSP34	Project Work – Phase 1	CSE	PW	0	0	6	6	6
Total					**6**	**0**	**11**	**17**	**17**

Where x=1,2,3… L –Lecture, T – Tutorial, P- Practical

Professional Elective Course VI*		**Professional Elective Course VII***	
Subject Code	**Subject Name**	**Subject Code**	**Subject Name**
MCSE311	Getting Started with Competitive Programming* https://nptel.ac.in/courses/106106231	MCSE321	Software Project Management* https://nptel.ac.in/courses/106105218
MCSE312	Social Networks* https://nptel.ac.in/courses/106106169	MCSE322	Computer System Optimizations for ML https://nptel.ac.in/courses/106108003
MCSE313	Responsible & Safe AI Systems* https://nptel.ac.in/courses/106106472	MCSE323	Generative AI for Computer Vision https://nptel.ac.in/courses/106103001

NOTE: * courses have to be completed using online NPETL platform. If course content is not available on NPTEL then recorded lectures to be provided to students. Evaluation for CIE and SEE to be done by faculty in-charge.

M.Tech. in Computer Science and Engineering Scheme of Teaching and Examination Batch 2026-28									
IV SEMESTER									
Sl. No.	**Subject Code**	**Subject**	**Teaching Department**	**Category**	**Credits**				**Total contact hours /week**
					L	**T**	**P**	**Total**	
1	MCSP41	Project Work – Phase II	CSE	PW	0	0	18	18	18
Total					**0**	**0**	**18**	**18**	**18**

L –Lecture, T – Tutorial, P- Practical

ADVANCED ENGINEERING MATHEMATICS	
Course Code: MCS11	**Credits: 2:1:0**
Pre-requisites: Engineering Mathematics I- IV	**Contact Hours: 28+28**
Course Coordinator: Dr. A. Sreevallabha Reddy	

Course Content

Unit I

Linear Algebra – I Vectors and Linear Combinations, Vector Spaces, The Null space of A, Solving Ax = 0. The Complete Solution to Ax = b, Independence, Basis and Dimension, Dimensions of the Four Subspaces, Orthogonality of the Four Subspaces, Projections. Orthonormal Bases and Gram-Schmidt Method, Factorization into A = QR, Least Squares Approximations.

Unit II

Linear Algebra – II: Linear Transformation: Fundamentals, The Matrices of a linear Transformation., Change of basis. Introduction to Eigenvalues and Eigenvectors, Similarity and Diagonalization. Symmetric Matrices, The singular value decomposition (SVD).

Unit III

Random Variables: Review of Random Variables, Probability Distributions:Binomial distribution, Poisson distribution, Normal distribution, Exponential distribution and Uniform distribution.

Unit IV

Joint Probability Distributions and Stochastic Process Joint probability distribution: Joint probability distribution (both discrete and continuous). Stochastic Processes: Introduction, Classification of stochastic processes, discrete time processes,Stationary, Ergodicity, Autocorrelation.

Unit V

Markov Process, Introduction to Queuing and Applications: Introduction, Markov chain and Transition probabilities, Continuous Parameter Markov Chain, M/M/1: ∞/FIFO, K/FIFO, M/G/1 Queuing system characteristics, Case studies.

Text Books:

1. Gilbert Strang: Linear Algebra and its Applications, 5th Edition (2016).
2. David C Lay: Linear Algebra and its Applications, 5th Edition (2015).
3. Sheldon M. Ross: Probability models for Computer Science, Academic Press, 2009.
4. B.S.Grewal, Higher Engineering Mathematics, Khanna Publishers, 40thedition, 2007.
5. R.E. Walpole, R. H. Myers, R. S. L. Myers and K. Ye: Probability and Statistics for Engineers and Scientists, Pearson Education, Delhi, 8th edition, 2007.

Reference Books:

1. Murray R Spiegel, John Schiller & R. Alu Srinivasan: Probability and Statistics, Schaum's Outlines, 2nd edition.
2. Kishor S. Trivedi: Probability & Statistics with Reliability, Queuing and Computer Science Applications, PHI, 2nd edition, 2002.
3. Garreth Williams: Linear Algebra with Applications, Jones and Bartlett Press, 4th edition, 2001.
4. Erwin Kreyszig: Advanced Engineering Mathematics, Wiley India,10th edition, 2015.

Course Outcomes (COs):

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

1. Solve the system of equations AX=B. (PO 1, 3)
2. Find SVD and PCA of the given matrix. (PO 1, 3)
3. Express the probability distribution arising in the study of engineering problems and their applications. (PO 1, 3)
4. Apply the Markov Chain in prediction of future events. (PO 1, 3)
5. Apply and calculate the various parameters of the queuing models. (PO 1, 3)

ADVANCED DATABASE MANAGEMENT SYSTEMS	
Course Code: MCS12	**Credits: 4:0:0**
Pre-requisites: DBMS	**Contact Hours: 56**
Course Coordinator: Dr. Ganeshayya Shidaganti	

Course Content

UNIT I

Introduction to Database Models: Overview of relational data model: Relational Model Concepts, Relational Model Constraints and Relational Database Schemas, Database Design using ER model. **The Enhanced Entity-Relationship (EER) Model:** Subclasses, Super classes, and Inheritance, Specialization and Generalization, Constraints and Characteristics of Specialization and Generalization Hierarchies, Modeling of UNION Types Using Categories, A Sample UNIVERSITY EER Schema, Design Choices, and Formal Definitions

UNIT II

Indexing and Hashing: Basic Concepts, Ordered Indices, B+-Tree Index Files, Static Hashing, Dynamic Hashing, Bitmap Indices, Index Definition in SQL. **Query Processing:** Overview, Measures of Query Cost, Selection Operation, Sorting, Join Operation, Evaluation of Expressions **Query Optimization:** Transformation of Relational Expressions, Estimating Statistics of Expression Results, Choice of Evaluation Plans.

UNIT III

Introduction to PostgreSQL: A Short History of PostgreSQL, The PostgreSQL Architecture, Data Access with PostgreSQL. **Accessing Your Data using psql:** Some Basic psql Commands, Simple SELECT Statements. **Data Interfacing:** Adding, Updating, Deleting Data. **Advanced Data Selection:** Aggregate Functions. **Functions, Stored Procedures and Triggers:** Built-in Functions, Procedural Languages, Anatomy of a Stored Procedure, Defining a Trigger Procedure.

UNIT IV

Parallel Databases: I/O Parallelism, Inter-query Parallelism, Intra-query Parallelism, Intra-operation Parallelism, Interoperation Parallelism, Query Optimization, Design of Parallel Systems, Parallelism on Multi-Core Processors **Distributed Databases:** Homogeneous and Heterogeneous Databases, Distributed Data Storage, Distributed Transactions, Commit Protocols, Concurrency Control in Distributed Databases, Availability, Distributed Query Processing, Heterogeneous Distributed Databases, Cloud-Based Databases, Directory Systems

UNIT V

Enhanced Data Models: Motivation, Time in Databases, Spatial and Geographic Data, Multimedia Databases, Mobility and Personal Databases. **Variety of NoSQL Databases:** Data Management with Distributed Databases, The CAP Theorem, ACID and BASE. Four Types of NoSQL Databases: Key-value pair databases, Document databases, Column family store databases, Graph databases

Text Books:

1. Ramez Elmasri, Shamkant B. Navathe, **Fundamentals of Database Systems**, Six Edition, Pearson Publications.
2. Abraham Silberschatz, Henry F. Korth, S. Sudarshan, **Database System Concepts**, Sixth Edition, McGraw Hill Education, 2013
3. Stones, R., & Matthew, N. (2005**). Beginning databases with PostgreSQL: From novice to professional** (2nd ed.). Apress. Springer Nature. (CH:01,04,06,07,10)
4. **NoSQL for Mere Mortals** by Released April 2015, Publisher(s): Addison-Wesley Professional, ISBN: 9780134029894

Reference books:

1. Raghu Ramakrishnan, Johanners Gehrke, **Database Management Systems**, Second Edition. McGraw-Hill Education.
2. Guy Harrison, **Next Generation Databases: NoSQLand Big Data**, Apress Berkeley, CA, 2015. https://doi.org/10.1007/978-1-4842-1329-2
3. M. Tamer Özsu, Patrick Valduriez, **Principles of Distributed Database Systems**, Springer Cham, 2020. https://doi.org/10.1007/978-3-030-26253-2
4. Sadalage, P. & Fowler, **NoSQL Distilled**: A Brief Guide to the Emerging World of Polyglot Persistence, Pearson Addision Wesley, 2012

Course Outcomes (COs):

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

1. Apply the Relational Model and EER model to Design Databases (PO1, 3,5)
2. Discuss the concepts of indexing, hashing, query processing, and query optimization techniques used for efficient data storage and retrieval (PO1, 3,5)
3. Explain PostgreSQL architecture, data operations, querying, functions, stored procedures, and triggers used in database management.
4. Describe the processing of the queries in distributed and parallel databases. (PO1,3,5)
5. Elucidate the enhanced data models and NoSQL databases. (PO1,2, 3, 5)

ADVANCED COMPUTATIONAL INTELLIGENCE	
Course Code: MCS13	**Credits: 4:0:0**
Pre-requisites: AI, ML, Python Programming	**Contact Hours: 56**
Course Coordinator: Dr. Rajarajeswari S, Dr. Sangeetha V	

Course Content

Unit I

Introduction to Computational Intelligence: Computational Intelligence Paradigms; **Introduction to Artificial Neural Networks with Keras:** From Biological to Artificial Neurons, Implementing MLPs with Keras, Fine-Tuning Neural Network Hyperparameters; **Training Deep Neural Networks:** The Vanishing/Exploding Gradients Problems , Reusing Pretrained Layers , Faster Optimizers , Avoiding Overfitting Through Regularization .

Unit II

Custom Models and Training with TensorFlow: A Quick Tour of Tensor Flow, Using TensorFlow like NumPy , Customizing Models and Training Algorithms , TensorFlow Functions and Graphs; **Loading and Preprocessing Data with Tensor Flow:** The Data API , The TF Record Format , Preprocessing the Input Features , TF Transform; **Deep Computer Vision Using Convolutional Neural Networks:** The Architecture of the Visual Cortex , Convolutional Layers, Pooling Layers , CNN Architectures

Unit III

Processing Sequences Using RNNs and CNNs: Recurrent Neurons and Layers , Training RNNs, Forecasting a Time Series , Handling Long Sequences; **Natural Language Processing with RNNs and Attention:** Generating Shakespearean Text Using a Character RNN , Sentiment Analysis , An Encoder–Decoder Network for Neural Machine Translation , Attention Mechanisms; **Representation Learning and Generative Learning Using Autoencoders and GANs :** Efficient Data Representations , Performing PCA with an Undercomplete Linear Autoencoder , Stacked Autoencoders , Convolutional Autoencoders , Recurrent Autoencoders, Denoising Autoencoders, Sparse Autoencoders, Variational Autoencoders, Generative Adversarial Networks

Unit IV

Introduction to Evolutionary Computation: Generic Evolutionary Algorithm, Representation – The Chromosome, Initial Population ,Fitness Function, Selection , Reproduction Operators; **Genetic Algorithms**: Canonical Genetic Algorithm, Crossover, Mutation, Control Parameters, Genetic Algorithm Variants; **Evolutionary Programming**; Basic Evolutionary Programming, Evolutionary Programming Operators, Strategy Parameters, Evolutionary Programming

Implementations; **Particle Swarm Optimization:** Basic Particle Swarm Optimization, Social Network Structures, Basic Variations, Basic PSO Parameters

Unit V

Ant Algorithms: Ant Colony Optimization Meta-Heuristic, Cemetery Organization and Brood Care , Division of Labor; **Fuzzy Sets:** Formal Definitions, Membership Functions, Fuzzy Operators, Fuzzy Set Characteristics, Fuzziness and Probability; **Fuzzy Logic and Reasoning:** Fuzzy Logic, Fuzzy Inferencing; **Fuzzy Controllers**: Components of Fuzzy Controllers, Fuzzy Controller Types

Text Books:

1. Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow - Concepts, Tools, and Techniques to Build Intelligent Systems by Aurélien Géron, 2nd Edition(2019)
2. Andries P. Engelbrecht, "Computational Intelligence: An Introduction", 2nd Edition, Wiley-ISTE, 2007.

Reference Books:

1. Stuart Russell, Peter Norvig, "Artificial Intelligence A modern Approach", 4th edition, Pearson Prentice Hall, 2023.
2. Deep Learning by Ian Goodfellow, Yoshua Bengio, and Aaron Courville Edition: Latest (2016; remains a core reference in deep learning)
3. Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play" by David Foster Edition: 2nd Edition (2022)
4. Natural Language Processing with Transformers" by Lewis Tunstall, Leandro von Werra, and Thomas Wolf Edition: 1st Edition (2022)
5. Genetic Algorithms in Search, Optimization, and Machine Learning" by David E. Goldberg Edition: Reissue Edition (2021)

Course Outcomes:

1. *Analyze* deep learning models by applying computational intelligence concepts and implementing neural network architectures using Keras. (PO-1 ,2,3,5)
2. *Design* deep learning models using TensorFlow and convolutional neural networks for computer vision applications. (PO-1 ,2,3,5)
3. *Develop* sequence-processing and generative deep learning models using RNNs, attention mechanisms, autoencoders, and GANs for various real-world applications. (PO-1 ,2,3,5)
4. *Apply* evolutionary computation techniques such as genetic algorithms, evolutionary programming, and particle swarm optimization to solve complex optimization problems. (PO-1 ,2,3,5)
5. *Evaluate* ant-based optimization techniques and fuzzy logic systems for their effectiveness in intelligent control and decision-making applications. (PO-1 ,2,3,5)

MANDATORY CREDIT COURSE

RESEARCH METHODOLOGY AND IPR	
Course Code: RMI16	**Credits: 3:0:0**
Pre-requisites: Nil	**Contact Hours: 42**
Course Coordinator:	

Course Content

Unit I

Introduction: Meaning of Research, Objectives of Engineering Research, and Motivation in Engineering Research, Types of Engineering Research, Finding and Solving a Worthwhile Problem. Ethics in Engineering Research, Ethics in Engineering Research Practice, Types of Research Misconduct, Ethical Issues Related to Authorship.

Unit II

Literature Review and Technical Reading, New and Existing Knowledge, Analysis and Synthesis of Prior Art Bibliographic Databases, Web of Science, Google and Google Scholar, Effective Search: The Way Forward Introduction to Technical Reading Conceptualizing Research, Critical and Creative Reading, Taking Notes While Reading, Reading Mathematics and Algorithms, Reading a Datasheet. Attributions and Citations: Giving Credit Wherever Due, Citations: Functions and Attributes, Impact of Title and Keywords on Citations, Knowledge Flow through Citation, Citing Datasets, Styles for Citations, Acknowledgments and Attributions, What Should Be Acknowledged, Acknowledgments in, Books Dissertations, Dedication or Acknowledgments.

Unit III

Building Intellectual Property Rights, Law of Patents, Fundamentals of Patent Law - Evolution of the patent system, Patentability Requirements; Patentable Subject Matter; Industrial Applicability/Utility; Novelty; Anticipation by publication; Anticipation by public knowledge and public use; Anticipation by public display; Anticipation by sale; Inventive Step/Non-Obviousness; Novelty Assessment; Inventive Step Assessment; Specification, Drafting of A Patent Specification - Introduction Patent Specification; Provisional Specification Complete Specification, Parts of the complete specification; Patent Procedure in India - PATENT PROCEDURE; Registration and Renewal fee payment; Patent Infringement - Infringement of a patent; Literal Infringement; Equivalence Infringement; Indirect Infringement; Defenses - Experiment - Research or

Education - Bolar Exemption- Government use- Patent ExhaustionPatent Misuse- Inequitable Conduct - Remedies- Injunction- Account of profits- Costs; International Patent Regimes - International Instruments; Paris Convention; TRIPS AGREEMENT; PCT; BUDAPEST TREATY, Patenting Biotechnology Inventions - Unique nature of Biotechnology; Patentability Requirements and Biotechnology Inventions; Patentable Subject Matter- USA- Europe- India; Patentability of Software Inventions -Patentability of Software Inventions in USA; Patentability of software inventions in Europe; Patentability of Software Inventions in India.

Unit IV

Law of Copyright and Designs, Understanding Copyright Law - Historical Overview – Justification For Copyright Law - The Natural Law Justification - The Economic Rationale of Copyright Clause, Basic Concepts Underlying copyright Law - Idea – Expression Dichotomy Originality / Creativity – Fixation Email dated 09082022 HB Term of Protection, Subject - Matter of Copyright - Literary Works - Dramatic Works - Musical Work - Artistic Works - Cinematograph Films and Sound recordings, Acquisition of Copyright in India, Rights of the Copyright Owner - Economic Rights - Moral Right or Droid Moral Right of Authorship or Paternity Rights - Rights against Distortion or Mutilation of the Original Works or Integrity Rights - Limitations - Limitations set under International Regime – Berne Convention - Rome Convention - Trips Agreement - Three Step Test, Infringement of Copyright -Transfer of copyright - License and Assignment - License and consent -Duration of a License Form and Content - Disputes in Respect of License -Types of Licenses - Exclusive and Non- Exclusive Licenses.

Unit V

Basic Principles of Design Rights - Justification for Protecting Designs - Historical Perspective - Features of Shape, configuration, Pattern or Ornament - or Composition of lines or colour - New or Original - Applied to an Article, Excluded Subject - Matter - Method or Principle of Construction - Features Dictated Solely by Function - Mechanical Device - Trademark, or Property Mark, or Artistic Work - immoral Designs and Designs Contrary to Public order–Rights of the Owner of Designs and Tests for Infringement. Assignment of Design Rights, Infringement of Designs. Case Studies on Patents. Case study of Curcuma (Turmeric) Patent, Case study of Neem Patent, Case study of Basmati patent, Case study of Apple Inc. v. Samsung ElectronicsCo., Ltd.

Text Book:

1. Dipankar Deb • Rajeeb Dey, Valentina E. Balas "Engineering Research Methodology", ISSN 1868- 4394 ISSN 1868-4408 (electronic), Intelligent Systems Reference Library, ISBN 978-981-13- 2946-3 ISBN 978-981-13- 2947-0 (eBook), https://doi.org/10.1007/978-981-13-2947-0

Reference Book:

1. David V. Thiel "Research Methods for Engineers" Cambridge University Press,978-1-107-03488- 4

Course Outcomes (COs):

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

CO1. To know the meaning of engineering research. (PO 1,2,3,5)

CO2. To know the procedure o f Literature Review and Technical Reading. (PO 1,2,3,5)

CO3. To know the fundamentals of patent laws and drafting procedure. (PO 1,2,3,5)

CO4. Understanding the copyright laws and subject matters of copyrights and designs(PO 1,2,3,5)

CO5. Understanding the basic principles of design rights. (PO 1,2,3,5)

LABORATORY COURSES

ADVANCED DATABASE LABORATORY	
Course Code: MCSL17	**Credits: 0:0:1**
Pre-requisites: SQL and NoSQL / Python and Java Programming	**Contact Hours: 28**
Course Coordinator: Dr. Ganeshayya Shidaganti	

Course Content

1. Introduction to Entity-Relationship (ER) Model
2. Design an Enhanced Entity-Relationship (EER) Model
3. Implement SQL Queries using DDL, DML and DCL Statements
4. Implementation of complex queries (Nested queries and multiple joins)
5. Introduction to triggers and procedures.
6. Exercises on triggers and procedures.
7. Introduction to PostgreSQL and Conceptual Overview
8. PostgreSQL Data Types and Table Design
9. Build Applications in PostgreSQL
10. Introduction NoSQL Databases: Mongo DB
11. Implement NoSQL Queries using CRUD Operations.
12. Building Modern Applications in Mongo DB using NoSQL.

References:

1. "**Database Management Systems**" by Ragh Ramakrishnan, Johanners Gehrke, Second Edition. McGraw-Hill Education.
2. "**Fundamentals of Database Systems**" by Ramez Elmasri, Shamkant B. Navathe, Sixth Edition, Pearson Publications.
3. **PostgreSQL: Up and Running**, 2nd Edition by Regina O. Obe and Leo S. published by O'Reilly Media,2015.
4. **NoSQL for Mere Mortals** by Released April 2015, Publisher(s): Addison-Wesley Professional ,ISBN: 9780134029894
5. Sadalage, P. & Fowler, **NoSQL Distilled**: A Brief Guide to the Emerging World of Polyglot Persistence, Pearson Addision Wesley, 2012
6. **Graph Databases**, 2nd Edition by Ian Robinson, Jim Webber, Emil Eifrem Released June 2015 Publisher(s): O'Reilly Media, Inc. ISBN: 9781491930892
7. https://www.postgresql.org/download/
8. Stones, R., & Matthew, N. (2005). **Beginning databases with PostgreSQL**: From novice to professional (2nd ed.). Apress. Springer Nature. (CH:01,04,06,07,10)

Course Outcomes (COs):

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

1. Demonstrate the use of Oracle DB , Mongo DB and PostgreSQL Operations.
2. Apply SQL and NoSLQ statements to Query Different Database Models.
3. Develop a Real time Database Application Using Oracle DB/ PostgreSQL/ MongoDB /Neo4j of student's choice.

APPLICATION DEVELOPMENT USING PYTHON PROGRAMMING LABORATORY	
Course Code: MCSL18	**Credits: 0:0:1**
Pre-requisites: Nil	**Contact Hours: 28**
Course Coordinator: Dr. Chandrika Prasad	

Course Contents

There shall be a 1/2 exercises conducted on each of the following topics.

1. **Python Basics, Control Structures and Functions, Strings, lists, list comprehensions:** Manipulate synthetic data records to categorize and process using control structures and string/list operations.
2. **Tuples and Dictionaries, Lambdas and Regular Expressions, Objects and Classes:** Store/query synthesised data records using immutable structures, pattern matching, and classes for text processing in a mock system.
3. **Data Management and Visualization:** Use control structures and string operations to generate and process synthetic datasets, possible lab can be based on implementing if-else functions with list comprehensions for categorizing values and visualizing distributions with Matplotlib histograms.
4. **File I/O and Exceptions with Lists and NumPy:** Handle file operations with list processing and numerical arrays; possible lab can be based on reading CSV files using comprehensions, applying try-except for errors, and normalizing data with NumPy for robust extraction.
5. **Linear Regression with Tuples and Matplotlib:** Predict trends using immutable data structures and plots; possible lab can be based on storing pairs as tuples in dicts, training linear models with scikit-learn, and plotting regression lines with Matplotlib for MSE/R^2 analysis.
6. **KNN Classification with Control Structures and NumPy:** Classify data via nearest neighbors with loops and arrays, possible lab can be based on for-loop labelling with thresholds, computing distances using NumPy, and evaluating KNN accuracy/confusion matrices.
7. **Decision Trees with Objects and Visualization:** Build classifiers with class-based nodes and charts, possible lab can be based on defining Decision Node classes for tree simulation, training scikit-learn trees, and plotting importance bars with Matplotlib.
8. **K-Means Clustering with Comprehensions and Files:** Group data using list generation and file handling; possible lab can be based on synthetic lists via comprehensions, saving clusters with exception handling, and visualizing K-Means scatters with NumPy/Pandas.
9. **Text Processing and Sentiment Analysis with Lambdas and Regex:** Clean and analyze text with anonymous functions and patterns; possible lab can be based on lambda preprocessing with re.sub/findall for tokenization, vectorizing with TF-IDF, and training Naive Bayes for sentiment evaluation.
10. **Association Rules with Image Manipulation**: Mine patterns and visualize with overlays, possible lab can be based on applying Apriori for itemsets/rules

(support/confidence/lift), interpreting productivity links, and using Pillow to annotate frequency charts.

11. **Rule-based Chatbots with Functions and GUIs:** Create interactive responders with loops and interfaces, possible lab can be based on while-loop functions with regex parsing, lambda mapping, and Entry/Text for pattern-matched queries and unmatched handling.
12. **ML Pipelines with Classes and Threads:** Streamline workflows using objects and parallelism, possible lab can be based on DataProcessor classes for scaling/imputation, threading preprocessing, and evaluating saved pipelines with Pandas/NumPy on new data.
13. **Model Deployment with GUIs and Networks:** Deploy models via interfaces and sockets, possible lab can be based on receiving input/display with validation, socket clients for connectivity, and threaded concurrent requests with mock routing.
14. **Neural Networks with Matplotlib and Django:** Train NNs and integrate with plots/web, possible lab can be based on Keras feed-forward models for classification, plotting accuracy/loss with Matplotlib, and Django form endpoints for prediction JSON responses.

Reference Books:

1. Campbell, J., Gries, P., Montojo, J., & Wilson, G. (2016): Practical programming: An Introduction to Computer Science using Python. Pragmatic Bookshelf, 2nd Edition, 2016.

Course Outcomes (COs):

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

1. Design and implement python programs utilizing control structures, functions, strings and built-in data structures in Python. (PO1,2,3,4,5)
2. Employ object-oriented programming, files and exception handling and Linked List for solving problems using Python. (PO1,2,3,4,5)
3. Design and implement the concepts of numerical programming, Data Visualization and GUI programming using Python. (PO1,2,3,4,5)

PROFESSIONAL ELECTIVE COURSES ELECTIVES - I

CRYPTOGRAPHY AND NETWORK SECURITY	
Course Code: MCSE141	**Credits: 3:0:0**
Pre-requisites: Computer Networks	**Contact Hours: 42**
Course Coordinator: Dr. Sangeetha V	

Course Contents

Unit I

Introduction - Security Attacks, Security Services, Security Mechanisms, Cryptography.

Symmetric Key Cryptography - Block ciphers and stream ciphers, DES, AES (design principles and modes of operation), Modes of operation: ECB, CBC, CFB, OFB, CTR, GCM, Message authentication codes (MACs), Cryptographic hash functions (SHA-2, SHA-3)

Unit II

Asymmetric Key Cryptography- Public key cryptography principles, RSA algorithm and security, Diffie–Hellman key exchange, Elliptic Curve Cryptography (ECC), Digital signatures and key management

Unit III

Cryptographic Protocols- Authentication protocols, Key distribution and management, Secure socket layer (SSL) and Transport Layer Security (TLS 1.3), Secure e-mail (PGP, S/MIME), Secure shell (SSH)

System and Network Security -Security at different layers, IPsec architecture and modes, Firewalls and intrusion detection systems, Virtual Private Networks (VPNs), Secure routing and DNS security (DNSSEC)

Unit IV

Applied Cryptography in Modern Systems -, Network Endpoint Security – Firewalls, Intrusion Detection Systems, Malicious Software, Distributed Denial of Service Attacks**. Cloud Security** - Cloud Computing, Cloud Security Concepts, Cloud Security Risks and Countermeasures, Cloud Security as a Service, An Open-Source Cloud Security Module. **Internet of Things (IoT) Security** - The Internet of Things, IoT Security Concepts and Objectives, An Open-Source IoT Security Module. Mobile and wireless security, Secure storage and disk encryption, Trusted Platform Module (TPM)

Web and Application Security - Web security threats (XSS, CSRF, SQL injection), HTTPS and web authentication mechanisms, OAuth 2.0 and OpenID Connect, Secure API design

Unit V

Distributed Security - Cryptographic foundations of blockchain, Hash chains and Merkle trees, Consensus mechanisms (PoW, PoS), Smart contracts security issues

Emerging Topics in Cryptography - Post-quantum cryptography (intro), Zero-knowledge proofs (overview), Privacy-preserving cryptographic techniques, Lightweight cryptography for constrained devices

Reference Books:

1. William Stallings: Cryptography and Network Security-principles and practices, 8th Edition, Pearson India Education Services, 2023.
2. Behrouz A. Forouzan, Debdeep Mukhopadhyay: Cryptography and Network Security, 3rd Edition, Special Indian Edition, Tata McGraw-Hill, 2015.
3. Cryptography, Network Security and Cyber Laws – Bernard Menezes, Ravinder Kumar, Cengage Learning, 2018.
4. Charles P. Pfleeger, Shari Lawrence Pfleeger, Jonathan Margulies: Security in computing, 5th Edition, Pearson India Education Services, 2015.

Course Outcomes (COs):

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

1. Analyze the key management issues involved in symmetric key cryptosystems with respect to public key Cryptosystems (PO-1,2,3,4).
2. Demonstrate the various key distribution and management schemes (PO-1,2,3,4)
3. Apply Cryptographic Hash Functions & and verify messages using well known signature generation and verification algorithms (PO-1,2,3,4).
4. Examine the working of various Network security protocols. (PO-1, 2,3,4).
5. Illustrate the different network strategies, Cloud and Internet of Things (IoT) Security strategy (PO-1,2,3,4)

ADVANCED DISTRIBUTED SYSTEMS	
Course Code: MCSE142	**Credits: 3:0:0**
Pre-requisites: Operating System and Computer Networks	**Contact Hours: 42**
Course Coordinator: Dr. Sangeetha V	

Course Contents

Unit I

Introduction to Large-Scale Peer-to-Peer Distributed Systems: Large-Scale distributed systems, Consequences of large-scale, Some large-scale distributed systems, Architectures of large scale distributed systems. **Design Principles of Large-Scale Distributed System:** Introduction to peer-to-peer systems, The peer-to-peer paradigms, Services on structured overlays, Building trust in P2Psystems.

Unit II

Peer-to-Peer Storage: Introduction , BitTorrent, Gnutella. **Large-Scale Peer-to-Peer Game Applications:** Introduction, Large-scale game applications: model and specific requirements, Overview of peer-to-peer overlays for large-scale game applications, Overlays for FPS games, Overlays for online life-simulation games. **Introduction to Distributed Embedded and Real-time Systems:** Distributed real-time embedded systems, Safety critical systems as examples of DRE systems, Design process of DRE systems.

Unit III

Scheduling in Distributed Real-Time Systems: Introduction, Generalities about real-time systems, Temporal correctness, WCRT of the tasks, WCRT of the messages. **Software Engineering for Adaptative Embedded Systems:** Introduction, Adaptation, an additional complexity factor, Theoretical aspects of adaptation management , Technical solutions for the design of adaptative embedded systems, An example of adaptative system from the robotic domain, Applying MDE techniques to the design of the robotic use-case, Exploitation of the models.

Unit IV

The Design of Aerospace Systems : Introduction, Flight software typical architecture, Traditional development methods and their limits, Modeling a software system using TASTE: philosophy , Common solutions , What TASTE specifically proposes, Modeling process and tools, Technology , Model transformations, The

TASTE run-time , Illustrating our process by designing heterogeneous systems , First user feedback and TASTE future .

Unit V

Introduction to Security Issues in Distributed Systems: Secure data exchange, Security in specific distributed systems. **Practical Security in Distributed Systems :** Confidentiality, Authentication , Availability and fault tolerance, Ensuring resource security, Result checking in distributed computations. **Enforcing Security with Cryptography :** Cryptography: from a general perspective , Symmetric encryption schemes , Prime numbers and public key cryptography.

Text Books:

1. Serge Haddad, Fabrice Kordon , Laurent Pautet , Laure Petrucci , "Distributed Systems: Design and Algorithms", 1st Edition, Wiley-ISTE, 2013.
2. Maarten Van Steen and A.S. Tanenbaum, "Distributed Systems", 4th edition, Pearson Prentice Hall , 2023.
3. Félix F. Ramos, Herwig Unger, Victor Larios, "Advanced Distributed Systems", Springer Verlag Berlin Heidelberg, 2000.

Course Outcomes (COs):

At the end of the course, a student should be able to:

1. Identify various large-scale distributed systems and understand their architectures.(PO-1 ,3, 4)
2. Design and analyze peer-to-peer storage systems, large-scale peer-to-peer game applications, and understanding their specific architectural models, performance requirements, and safety considerations .(PO-1 ,3, 4)
3. Evaluate scheduling mechanisms for distributed real-time systems and apply software engineering principles to develop adaptive embedded systems. (PO-1 ,3, 4)
4. Model aerospace systems using the TASTE toolchain, applying model-based engineering principles to create reliable, safety-critical flight software. (PO-1 ,3, 4)
5. Apply cryptographic techniques to fault tolerance effectively to enforce security in real-world distributed applications. (PO-1 ,3, 4)

EXTENDED REALITY SYSTEMS	
Course Code: MCSE143	**Credits: 3:0:0**
Pre-requisites: Computer Graphics / Programming Fundamentals	**Contact Hours: 42**
Course Coordinator: Dr. Rajarajeswari S	

Course Content

UNIT – I

Introduction to VR and XR Foundations: Definition of Virtual Reality, Augmented Reality, Mixed Reality and Extended Reality; characteristics of immersive systems; modern VR/**XR** experiences; evolution of immersive technologies; bird's-eye view of XR systems: hardware, software, displays, sensors, interaction devices and development platforms.

The Geometry of Virtual and Augmented Worlds: geometric models, coordinate systems, position and orientation, axis-angle representation of rotation, homogeneous transformations, viewing transformations, chaining of transformations, stereo viewing, spatial registration and coordinate alignment in AR/MR environments.

Applications: education, training, healthcare, engineering visualization, simulation, digital twins and entertainment**.**

UNIT – II

Light, Optics, Displays and Human Perception in XR: Basic behavior of light, lenses, optical aberrations, field of view, resolution, refresh rate and latency. Display technologies for VR and XR: head-mounted displays, optical see-through displays, video see-through displays, projection-based displays, handheld AR and spatial computing devices.

Physiology of human vision: structure of the eye, eye movements, depth cues, vergence-accommodation conflict, visual fatigue and implications for XR design. Visual perception: perception of depth, motion and color; combining visual, auditory and vestibular information in immersive environments. Perceptual issues in AR/MR: real-virtual occlusion, lighting consistency, visual realism and user comfort.

UNIT III

XR Rendering, 3D Content Creation and Real-Time Performance: Visual rendering: ray tracing, rasterization, shading models, texture mapping, lighting models, shadows and physically based rendering. Stereo rendering, lens distortion correction, foveated rendering, frame rate optimization, latency reduction and motion-to-photon delay.

Rendering for XR: AR rendering pipeline, camera background rendering, depth-based occlusion, environmental lighting, anchors, plane detection and spatial mapping. Immersive photos and videos, 360-degree media, volumetric content and 3D asset optimization for XR.

Introduction to XR content creation using Blender and Unity: 3D modeling,

texturing, importing assets, scene creation, prefabs, materials, animation basics and optimization for real-time immersive applications.

UNIT – IV

Tracking, Interaction and Spatial Computing: Tracking: 2D orientation tracking, 3D orientation tracking, position and orientation tracking, inertial tracking, optical tracking, marker-based and markerless tracking. Inside-out and outside-in tracking, sensor fusion, SLAM, spatial anchors, environment scanning and 3D reconstruction. Tracking attached bodies, hand tracking, eye tracking, body tracking and controller tracking.

Interaction in VR/XR: motor programs and remapping, selection, manipulation, locomotion, teleportation, gesture-based interaction, gaze-based interaction, hand interaction, voice interaction and multimodal interaction. Social interaction in virtual and mixed environments, collaborative XR, avatars and shared spaces.

Development tools and standards: Unity XR Interaction Toolkit, AR Foundation, OpenXR, WebXR basics and cross-platform XR deployment considerations.

UNIT – V

XR Audio, Haptics, Evaluation, Safety and Future Directions: Audio in XR: physics of sound, auditory perception, spatial audio, binaural rendering, HRTF, sound localization and auditory cues for immersion. Haptic and somatosensory perception: two-point acuity, texture perception, force feedback, vibrotactile feedback, mid-air haptics and haptic interfaces.

Evaluation of VR/XR systems and experiences: usability, presence, immersion, task performance, simulator sickness, cybersickness, comfort, perceptual training and user studies. Recommendations for XR developers, accessibility, ethics, privacy, safety and responsible XR design.

Frontiers in XR: spatial computing, metaverse platforms, digital twins, robotic interfaces, brain–machine interfaces, AI-assisted XR content generation, XR for healthcare, education, defense, smart cities and Industry 4.0.

Textbooks:

1. Steven M. LaValle, Virtual Reality, Cambridge University Press, 2016.
2. Ralf Doerner, Wolfgang Broll, Paul Grimm and Bernhard Jung, Virtual and Augmented Reality (VR/AR): Foundations and Methods of Extended Realities (XR), Springer Cham, 2022.
3. Dieter Schmalstieg and Tobias Höllerer, Augmented Reality: Principles and Practice, Addison-Wesley Professional, 2016.
4. Anna Braun and Raffael Rizzo, XR Development with Unity: A Beginner's Guide to Creating Virtual, Augmented, and Mixed Reality Experiences Using Unity, Packt Publishing, 2024.
5. Roland Hess, Blender Foundations: The Essential Guide to Learning Blender, Focal Press

References:

1. William R. Sherman and Alan B. Craig, Understanding Virtual Reality: Interface, Application, and Design, 2nd Edition, Morgan Kaufmann / Elsevier, 2018.
2. Jesse Glover and Jonathan Linowes, Complete Virtual Reality and Augmented Reality Development with Unity, Packt Publishing, 2019.
3. Franz Lanzinger, 3D Game Development with Unity, 1st Edition, CRC Press / Routledge, 2022.
4. Lem Apperson, Unity 6 Game Development with C# Scripting: Leverage C# Scripting in Unity to Create Immersive Games and VR Experiences, Packt Publishing, 2025.
5. Jonathan Linowes, Unity Virtual Reality Projects, Packt Publishing.
6. Alan B. Craig, Understanding Augmented Reality: Concepts and Applications, Morgan Kaufmann, 2013.
7. Unity Technologies, **Create Virtual and Mixed Reality Experiences in Unity**, Unity Technical Guide / e-Book.
8. Apple Developer Documentation, **visionOS and Spatial Computing Development Resources**.
9. Khronos Group, **OpenXR Specification and Developer Resources**.

Course Outcomes (COs):

At the end of the course the student will be able to:

1. Understand the fundamentals of VR, AR, MR, and XR**,** including their architectures, geometric foundations, and real-world applications.
2. Explain the principles of light, optics, display technologies, and human perception to design effective and comfortable immersive XR experiences
3. Apply XR rendering techniques and 3D content creation tools to develop optimized, real-time immersive virtual and augmented environments
4. Analyze tracking, interaction techniques, spatial computing, and XR development frameworks for building interactive and collaborative XR applications
5. Evaluate XR systems using audio, haptics, usability, safety, ethics, and emerging technologies to design responsible and future-ready immersive applications.

PROFESSIONAL ELECTIVE COURSES ELECTIVES - II

DATA SCIENCE AND MANAGEMENT	
Course Code: MCSE151	**Credits: 3:0:0**
Pre-requisites: Nil	**Contact Hours: 42**
Course Coordinator: Dr. Rajarajeswari S	

Course Contents

Unit 1

Introduction to Data Science and R Tool, Overview of Data Science Importance of Data Science in Engineering, Data Science Process, Data Types and Structures, Introduction to R Programming, Basic Data Manipulation in R, Simple programs using R. Introduction to RDBMS: Definition and Purpose of RDBMS Key Concepts: Tables, Rows, Columns, and Relationships, SQL Basics: SELECT, INSERT, UPDATE, DELETE Importance of RDBMS in Data Management for Data Science

Unit 2

Linear Algebra for Data Science, Algebraic View, Vectors and Matrices, Product of Matrix & Vector, Rank and Null Space, Solutions of Over determined Equations, Pseudo inverse, Geometric View, Vectors and Distances, Projections, Eigenvalue Decomposition.

Unit 3

Statistical Foundations, Descriptive Statistics, Notion of Probability, Probability Distributions Understanding Univariate and Multivariate Normal Distributions, Mean, Variance, Covariance, and Covariance Matrix, Introduction to Hypothesis Testing, Confidence Intervals for Estimates.

Unit 4

Optimization and Data Science Problem Solving, Introduction to Optimization Understanding Optimization Techniques, Typology of Data Science Problems, Solution Framework for Data Science Problems.

Unit 5

Regression and Classification Techniques, Linear Regression, Simple Linear Regression and Assumptions, Multivariate Linear Regression, Model Assessment and Variable Importance, Subset Selection, Classification Techniques, Classification using Logistic Regression.

Text Books:

1. "Python for Data Analysis" by Wes McKinney, 2nd Edition (2018)
2. "Data Science from Scratch: First Principles with Python" by Joel Grus, 2nd Edition (2019)

Reference Books:

1. "An Introduction to Statistical Learning" by Gareth James, Daniela Witten, Trevor Hastie, and Robert Toshigami, 2nd Edition (2021)
2. "The Elements of Statistical Learning" by Trevor Hastie, Robert Toshigami, and Jerome Friedman, 2nd Edition (2009)

3. "Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking" by Foster Provost and Tom Fawcett, 2nd Edition (2013)
Web links and Video Lectures (e-Resources):
https://www.coursera.org/specializations/jhu-data-science
https://www.kaggle.com/learn/data-science

Course Outcomes (COs):

At the end of the course, a student should be able to:

1. Explain the fundamental concepts of data science, data management, R programming, and relational database systems for data-driven applications. (PO1, 3,4)
2. Apply linear algebra concepts, including vectors, matrices, eigenvalue decomposition, and pseudo-inverse, in data science problem solving. (PO1, 2, 3, 4)
3. Analyze datasets using statistical methods, probability distributions, hypothesis testing, and confidence intervals. (PO1, 2, 3, 4)
4. Apply optimization techniques and data science problem-solving frameworks to model and solve engineering problems. (PO1, 2, 3, 4,5)
5. Develop regression and classification models using statistical and machine learning techniques for predictive data analysis. (PO1, 2, 3, 4,5)

IoT AND EDGE AI	
Course Code: MCSE152	**Credits: 3:0:0**
Pre-requisites: Nil	**Contact Hours: 42**
Course Coordinator: Mrs. Veena G S	

Course Contents

UNIT–I: Internet of Things – Fundamentals and Architecture

Definition, evolution, and vision of Internet of Things, Characteristics, constraints, and challenges of IoT systems, IoT components: sensors, actuators, gateways, and cloud, IoT architectural models: Three-layer and five-layer architectures,Device–Gateway–Cloud architecture,IoT design methodology and use-case analysis, IoT application domains: smart cities, healthcare, agriculture, Industry 4.0

UNIT–II: IoT Devices, Embedded Systems, and Communication Protocols

Sensors and actuators: types, characteristics, and interfacing, Embedded platforms for IoT: Arduino, ESP32, Raspberry Pi (overview), Real-Time Operating Systems (RTOS): RTOS concepts, FreeRTOS and Zephyr overview, IoT networking technologies: Bluetooth Low Energy, ZigBee, Wi-Fi, LoRaWAN, NB-IoT, Application layer protocols: MQTT, CoAP, AMQP, HTTP/REST, IPv6 and 6LoWPAN fundamentals

UNIT–III: Edge and Fog Computing Architectures

Motivation for edge computing,Edge computing vs fog computing vs cloud computing, Edge computing architectural models:Device edge, gateway edge, and cloud edge

Resource constraints and latency considerations at the edge,Virtualization at the edge: Containers and lightweight virtualization, Orchestration concepts: Kubernetes fundamentals, K3s overview,Quality of Service (QoS) and performance considerations

UNIT–IV: Data Analytics and Machine Learning at the Edge

Characteristics of IoT data: streaming and time-series data, Data acquisition and preprocessing at the edge, Edge analytics pipelines, Machine learning techniques for IoT: Classification, regression, anomaly detection, TinyML and Edge AI:TensorFlow Lite, model quantization and optimization,Edge-based computer vision:OpenCV fundamentals, lightweight deep learning models,Case studies: predictive maintenance and smart surveillance

UNIT–V: Security, Platforms, and Emerging Trends

Security challenges and threat models in IoT and edge environments,Cryptographic mechanisms:Symmetric and asymmetric cryptography, TLS/DTLS, Secure boot, device authentication, and access control,IoT and edge platforms: AWS IoT Core, Azure IoT HubAWS Greengrass, Azure IoT Edge,Emerging trends:Digital twins,Blockchain for IoT,Zero Trust IoT architectures, Industrial IoT (IIoT) and Industry 4.0

Reference Books

1. Bahga, A., & Madisetti, V., Internet of Things: A Hands-On Approach, VPT
2. Hanes, D., Salgueiro, G., Grossetete, P., IoT Fundamentals, Cisco Press
3. Shi, W. et al., Edge Computing: Vision and Challenges, IEEE Press
4. Raj Kamal, Internet of Things – Architecture and Design, McGraw Hill
5. McEwen, A., & Cassimally, H., Designing the Internet of Things, Wiley
6. Pete Warden & Daniel Situnayake, TinyML, O'Reilly
7. Buyya, R., Mastering Cloud Computing, McGraw Hill
8. Alla, S., Blockchain and IoT, Packt

Course Outcomes (COs)

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

1. Explain IoT architectures, components, devices, and communication models used in smart connected systems (PO1, 3, 4)
2. Design IoT systems using appropriate devices and communication protocols (PO1, 3, 4, 5)
3. Analyze and implement edge and fog computing architectures for edge-enabled applications (PO1, 2, 3, 4, 5)
4. Apply edge analytics and TinyML techniques for intelligent IoT applications using lightweight machine learning models. (PO1, 2, 3, 4, 5)
5. Analyze security mechanisms, cloud-edge platforms, and emerging trends in IoT and Edge AI ecosystems. (PO1, 2, 3, 4, 5)

IMAGE PROCESSING AND MACHINE VISION	
Course Code: MCSE153	**Credits: 3:0:0**
Pre-requisites: Nil	**Contact Hours: 42**
Course Coordinator: Dr. T N R Kumar	

Course Content

Unit I

Introduction and Digital Image Fundamentals, Motivation & Perspective,Applications, Components of Image Processing System, Fundamentals Steps in Image Processing, Image Sampling and Quantization, Some basic relationships like Neighbors, Connectivity, Distance Measures between pixels

Unit II

Image Enhancement in the Spatial and Frequency Domain: Image enhancement by point processing, Image enhancement by neighborhood processing, Basic Gray Level 20% Transformations, Histogram Processing, Enhancement Using Arithmetic and Logic operations, Zooming, Basics of Spatial Filters, Smoothening and Sharpening Spatial Filters, Combining Spatial Enhancement Methods. Introduction to FourierTransform and the frequency Domain, Smoothing and Sharpening Frequency Domain Filters, Homomorphic Filtering

Unit III

Image Restoration and Image Compression, Model of The Image Degradation / Restoration Process, Noise Models, Restoration in the presence of Noise Only Spatial Filtering, Periodic Noise Reduction by Frequency Domain Filtering, Linear Position-Invariant Degradations, Estimation of Degradation Function, Inverse filtering, Wiener filtering, Constrained Least Square Filtering, Geometric Mean Filter, Geometric Transformations. Data Redundancies, Image Compression models, Elements of Information Theory, Lossless and Lossy compression, Huffman Coding, Shanon-Fano Coding, Arithmetic Coding, Golomb Coding, LZW Coding, Run Length Coding, Loss less predictive Coding, Bit Plane Coding, Image compression standards.

Unit IV

Image Segmentation and Morphological Image Processing, Discontinuity based segmentation, similarity based segmentation, Edge linking and boundary detection, 20% Threshold, Region based Segmentation Introduction to Morphology, Dilation, Erosion, Some basic Morphological Algorithms

Unit V

Object Representation and description and Computer Vision Techniques
Introduction to Morphology, Some basic Morphological Algorithms, Representation, Boundary Descriptors, Regional Descriptors, Chain Code, Structural Methods. Review of Computer Vision applications; Fuzzy-Neural algorithms for computer vision applications

Text Books:

1. Rafael C. Gonzalez & Richard E. Woods: Digital Image Processing, Pearson Education, 4th edition, 2018.
2. David Forsyth, Jean Ponce, Computer Vision: A Modern Approach,Pearson Education, 2nd edition, 2012.

Reference Books:

1. Wilhelm Burger, Mark J. Burge, Digital Image Processing: An Algorithmic Introduction, Springer; 3rd ed. 2022 edition
2. A.K. Jain, Fundamentals of Digital Image Processing, PHI
3. Richard Szeliski, Computer Vision: Algorithms and Applications, Springer; 2nd ed. 2021
4. W.K. Pratt, Digital Image Processing, John Wiley & Sons; 4th Edition, 2007.

Course Outcomes (COs):

At the end of the course the student will be able to:

1. Explain the fundamentals of image processing and computer vision. (PO1,3,4)
2. Illustrate the image enhancement techniques. (PO1,3,4)
3. Illustrate Image restoration and image compression technique. (PO1,3,4)
4. Explain image segmentation and morphological image processing. (PO1,3,4)
5. Summarize computer vision techniques and its uses. (PO1,3,4)

SEMESTER II

ADVANCED DATA STRUCTURES AND ALGORITHMS	
Course Code: MCS21	**Credits: 3:1:0**
Pre-requisites: Analysis and Design of Algorithm	**Contact Hours: 42+28**
Course Coordinator: Dr. Sangeetha J	

Course Content

Unit I

Analysis Techniques: Growth of Functions - Asymptotic notation, Standard notations and common functions.

Divide-and-Conquer: The maximum-subarray problem, Strassen's algorithm for matrix multiplication, The substitution method for solving recurrences, The recursion-tree method for solving recurrences, The master method for solving recurrence, Proof of the master theorem.

Amortized Analysis: Aggregate, The Accounting method, The Potential Methods, Dynamic tables.

Unit II

Advanced Data Structures - B-Trees - Definition of B-trees, Basic operations on B-trees, Deleting a key from a B-tree, Red-Black trees - Properties of red-black trees, Rotations, Insertion, Deletion. Hashing: Hash Tables, Direct-address tables, Hash tables, Hash functions, open addressing, perfect hashing

Heaps: Binomial Heaps, Fibonacci Heaps: Structure of Fibonacci heaps, Mergeable-heap operations, Decreasing a key and deleting a node.

Dynamic Programming: Rod cutting, Matrix-chain multiplication, Elements of dynamic programming, Longest common subsequence, Optimal binary search trees.

Unit III

Greedy Algorithms: An activity-selection problem, Elements of the greedy strategy, Huffman codes, Matroids and greedy methods, A task-scheduling problem as a matroid.

Minimum Spanning Trees: Growing a minimum spanning tree, The algorithms of Kruskal and Prim

Single-Source Shortest Paths: The Bellman-Ford algorithm, Single-source shortest paths in directed acyclic graphs, Dijkstra's algorithm, Difference constraints and shortest paths.

Unit IV

All-Pairs Shortest Paths: Shortest paths and matrix multiplication, The Floyd-Warshall algorithm, Johnson's algorithm for sparse graphs

Maximum Flow: Flow networks, The Ford-Fulkerson method, Maximum bipartite matching, Push-relabel algorithms

String Matching: The naive string-matching algorithm, The Rabin-Karp algorithm, String matching with finite automata, The Knuth-Morris-Pratt algorithm

Unit V

Number Theoretic Algorithms: Elementary notations, GCD, Modular Arithmetic, The Chinese remainder theorem, RSA cryptosystem.

NP-Completeness: Polynomial time, Polynomial-time verification, NP-completeness and reducibility, NP-completeness proofs, NP-complete problems.

Approximation Algorithms: The vertex-cover problem, The traveling-salesman problem, The set-covering problem, Randomization and linear programming, The subset-sum problem

Algorithmic Puzzles: Magic Square, n-queen's problem, Glove Selection, Ferrying Soldiers, Jigsaw Puzzle Assembly, A Stack of Fake Coins, Maximum Sum Descent, Hats of Two Colors, Pluses and Minuses, searching for a Pattern, Locker Doors, Palindrome Counting, inverting a Coin Triangle, Sorting 5 in 7.

Tutorial:

Lesson No.	Topic	No. of Hours
	Tutorials	**14**
1.	Problems on Asymptotic functions and bounds	02
2.	Problems on Recurrence Relations - substitution method, The recursion-tree method	02
3.	Problems on Recurrence Relations - The master method	02
4.	Problems on Divide and Conquer Techniques: Strassen's algorithm for matrix multiplication	02
5.	Problems based on Dynamic Programming: Matrix-chain multiplication	02
6.	Problems based on Greedy Techniques: An activity-selection problem	02
7.	Problems based on Greedy Techniques: A task-scheduling problem as a matroid.	02
8.	Problems based on Single-Source Shortest Path: Dijkstra's algorithm	02
9.	Problems based on All-Pairs Shortest Paths: Johnson's algorithm for sparse graphs	02
10.	Problems based on Maximum Flow: Maximum bipartite matching	02
11.	Problems based on Maximum Flow: Ford-Fulkerson method	02
12.	Problems based on String Matching: Rabin-Karp algorithm	02
13.	Problems on Approximation Algorithms: The set-covering problem	02

14.	Problems on Approximation Algorithms: The vertex-cover problem	02

Text Books:

1. T H Cormen, C E Leiserson, R L Rivest and C Stein: Introduction to Algorithms, 3/e, PHI, 2011.
2. Anany Levitin and Maria Levitin: Algorithmic Puzzle, Oxford University Press, 2011
3. Mark Allen Weiss: Data Structures and Algorithm Analysis in C++, 3rd Edition, Pearson Education, 2011.

Reference Books:

1. Introduction to the Design & Analysis of Algorithms - Anany Levitin, 2nd Edition, Pearson Education, 2007.
2. Algorithm Design - Jon Kleinberg and Eva Tardos, Pearson, 1st Edition (2013).

Course Outcomes (COs):

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

1. Demonstrate the ability to analyze algorithmic problems using asymptotic notation, recurrence relations, and amortized analysis techniques to evaluate performance and optimize solutions. (PO1,2,3,4,5)
2. Design, implement, and apply advanced data structures such as B-trees, red-black trees, hash tables, and Fibonacci heaps to solve complex computational problems efficiently and different dynamic programming techniques. (PO1,2,3,4,5)
3. Apply greedy strategies, and graph-based algorithms to solve optimization problems, including shortest path, spanning tree, and scheduling challenges. (PO1,2,3,4,5)
4. Solve problems involving maximum flow, string matching, and all-pairs shortest paths using advanced algorithms like Ford-Fulkerson, Knuth-Morris-Pratt, and Floyd-Warshall, and adapt these methods to real-world applications. (PO1,2,3,4,5)
5. Analyze the computational complexity of problems, prove NP-completeness, and design approximation algorithms for intractable problems, while exploring algorithmic puzzles to foster creative and innovative thinking. (PO1,2,3,4,5)

ADVANCED OPERATING SYSTEMS	
Course Code: MCS22	**Credits: 4:0:0**
Pre-requisites: **Operating Systems**	**Contact Hours: 56**
Course Coordinator: **Dr. Chandrika Prasad**	

Course Content

Unit I

Introduction to the Linux Kernel- Obtaining the Kernel Source, The Kernel Source Tree, Building the Kernel. Debugging - Getting Started, Bugs in the Kernel, Debugging by Printing, Oops, Kernel Debugging Options. **Process Management** – process descriptor and task structure, process creation, The Linux Implementation of Threads, process termination.

Unit II

Process Scheduling – Multitasking, Linux's Process Scheduler, Policy, Linux scheduling algorithm, The Linux Scheduling Implementation, Preemption and Context Switching, Real-Time Scheduling Policies, Scheduler-Related System Calls

Unit III

Kernel Data Structures - What Data Structure to Use, When, **Interrupt Handlers** - Top Halves Versus Bottom Halves, Registering an Interrupt Handler. **Bottom Halves** - softirq, tasklets, Work Queues; **Timer**- Jiffies, Hardware Clocks and Timers

Unit IV

Memory Management – Pages, Zones, Getting Pages, kmalloc, vmalloc, slab layer, **The Process Address Space-** Address Spaces, memory descriptor. **The Virtual Filesystem** - Common Filesystem Interface, Filesystem Abstraction Layer, **Unix Filesystems**, **VFS Objects** and Their Data Structures

Unit V

Dockers-Docker Basics and Architecture, what is containerization, how are containers different from **Dockers-**Docker Basics and Architecture, what is containerization, how are containers different from physical machines and VMs, Docker evolution and architecture, Developments in Docker world, Docker tooling, Basic Docker commands. **Docker Networking-** Introduction, Types of Docker networks, Using Networks, Identifying container networks

ReferenceBooks:

1. Billimoria, Kaiwan N. Linux Kernel Programming: A comprehensive guide to kernel internals, writing kernel modules, and kernel synchronization. Packt Publishing Ltd, 2021.
2. BharadwaJ, Raghu. Mastering Linux Kernel Development: A kernel developer's reference manual. Packt Publishing Ltd, 2017.
3. Robert Love, Linux kernel development, 3rd edition, 2010**.**

4. Farhan Hasin Chowdhury, The Docker Handbook – Learn Docker for Beginners, https://www.freecodecamp.org/news/the-docker-handbook/, freeCodeCamp press, 2021.

CourseOutcomes (COs):

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

1. Examine the kernel components for building kernel image from kernel source (PO-1,2,3)
2. Analyze kernel process and CPU scheduling to observe the process management (PO-1,2,3,5)
3. To understand the working of the interrupt handler (PO-1,2,3,5)
4. Demonstrate memory management at kernel level (PO-1,2,3,5)
5. Illustrate Dockers techniques for various networking and file system applications (PO1, 2, 3,5).

CLOUD COMPUTING AND MICROSERVICES	
Course Code: MCS23	**Credits: 4:0:0**
Pre-requisites: Nil	**Contact Hours: 56**
Course Coordinator: Dr. J Geetha	

Course Content

Unit I

Cloud Computing Foundations: Evolution and Characteristics of Cloud Computing, Service Models: IaaS, PaaS, SaaS, Deployment Models: Public, Private, Hybrid, Community, Virtualization: Hypervisors, Containers vs VMs, Introduction to major cloud platforms: AWS, Azure, GCP.

Pedagogy / Course delivery tools: Chalk and talk, Power point presentation, Videos.

Lab Component / Practical Topics: Create an EC2/VM instance, explore cloud dashboards, Compare VMs vs Containers

Unit II

Cloud Infrastructure and Services: Compute, Storage, and Networking in Cloud, Object Storage (S3/GCS), Block and File Storage, Load Balancers, Auto-scaling Serverless Computing (AWS Lambda, GCP Cloud Functions) Cloud Databases: RDS, DynamoDB, Firebase.

Pedagogy / Course delivery tools: Chalk and talk, Power point presentation, Videos.

Lab Component / Practical Topics: Upload/download data to S3/GCS, - Launch a serverless function (Lambda), Use Load Balancer & Auto-scaling demo

Unit III

Microservices Architecture Design: Monolithic vs Microservices Architecture, Principles of Microservices: SRP, Bounded Contexts, Domain-Driven Design, Inter-service Communication: REST, gRPC ,API Gateway, Service Discovery, Circuit Breaker Pattern Versioning and Service Contracts

Pedagogy / Course delivery tools: Chalk and talk, Power point presentation, Videos.

Lab Component / Practical Topics: Create 2 RESTful microservices (e.g., product & order), Implement API Gateway using NGINX/Postman, Inter-service communication demo

Unit IV

Containers and Orchestration: Docker: Images, Containers, Volumes, Docker Compose,Kubernetes Basics: Pods, Deployments, Services, ConfigMaps Container Networking and Service Exposure,Cloud-native deployment using Kubernetes,Intro to Service Mesh and Sidecars.

Pedagogy / Course delivery tools: Chalk and talk, Power point presentation, Videos.

Lab Component / Practical Topics: Build Docker images, Use Docker Compose for multi-container apps, Deploy app on Minikube/K8s cluster

Unit V

Cloud-Native Design and 12-Factor App: introduction to Cloud-Native Applications,12-Factor App Methodology: Best practices for building scalable, maintainable microservices,Stateless Processes, Config via Environment, Port Binding, Logs, Admin Tasks, Observability in Microservices: Logging, Monitoring, and Tracing, Case Studies: Netflix, Uber, Amazon Microservices Patterns,Cloud-Native Deployment and Dev/Prod Parity using Containers

Pedagogy / Course delivery tools: Chalk and talk, Power point presentation, Videos.

Lab Component / Practical Topics: - Implement 6 of the 12-Factor principles (env config, logs, port binding, etc.), - Monitor microservice using Prometheus + Grafana, - Use .env and logging best practices

Text books:

1. Cloud Computing: Concepts, Technology & Architecture – Thomas Erl
2. Building Microservices – Sam Newman
3. Cloud Native DevOps with Kubernetes – John Arundel, Justin Domingus
4. The Twelve-Factor App – https://12factor.net (Online Reference)

Reference Books/Links:

1. Marinescu, Dan C. **Cloud computing: theory and practice**. Morgan Kaufmann, 2022. – 3rd Edition – Elsevier.
2. White, Tom. **Hadoop: The definitive guide**. " O'Reilly Media, Inc.", 2012.

Third Edition.
3. Ryza, Sandy, Uri Laserson, Sean Owen, and Josh Wills. **Advanced analytics with spark: patterns for learning from data at scale**. " O'Reilly Media, Inc.", 2017. 2nd Edition,
4. Distributed and Cloud Computing, From Parallel Processing to the Internet of Things, Kai Hwang, Jack Dongarra, Geoffrey Fox. MK Publishers.
5. Cloud Computing: A Practical Approach, Anthony T. Velte, Toby J. Velte, Robert Elsenpeter, McGraw Fill, 2010.
6. Cloud Computing A hands - on approach – Arshdeep Bahga & Vijay madisetti Universities press.
7. https://docs.rightscale.com/cm/designers_guide/cm-cloud-computing-system-architecture-diagrams.htm

Course Outcomes (COs):

At the end of the course, the student should be able to:

1. Understand the fundamentals of cloud computing, including service models, deployment models, virtualization technologies, and major cloud platforms.
2. Apply cloud infrastructure services such as compute, storage, networking, serverless computing, and cloud databases to build scalable cloud solutions.
3. Design microservices-based applications using architectural principles, service communication mechanisms, API gateways, and resilience patterns.
4. Develop and deploy containerized applications using Docker, Kubernetes, and cloud-native orchestration techniques.
5. Evaluate cloud-native applications using the 12-Factor methodology, observability tools, and industry best practices for building reliable, scalable, and maintainable systems.

LABORATORY COURSES

CLOUD COMPUTING AND MICROSERVICES LABORATORY	
Course Code: MCSL27	**Credits: 0:0:1**
Pre-requisites: Nil	**Contact Hours: 28**
Course Coordinator: Dr. Ganeshayya Shidaganti	

Course Content

List of problems for which student should develop program and execute in the Laboratory

1. Familiarization with AWS/Azure/GCP cloud platform. Create an account, explore cloud dashboard, and understand cloud resources.
2. Launch and configure a Virtual Machine (AWS EC2/Azure VM/GCP Compute Engine). Connect using SSH/RDP and perform basic administration tasks.
3. Compare Virtual Machines and Containers. Install Docker and execute basic container lifecycle commands.
4. Explore cloud storage services. Create Object Storage (S3/GCS/Azure Blob), upload/download files, configure permissions, and compare object, block, and file storage.
5. Implement Serverless Computing by creating a simple AWS Lambda (or Google Cloud Function/Azure Function) triggered through an HTTP request.
6. Configure Load Balancer and Auto Scaling concepts using cloud services or simulation. Analyze scalability and fault tolerance.
7. Design and implement two RESTful microservices (e.g., Product Service and Order Service) using Spring Boot/Node.js/Python Flask.
8. Implement inter-service communication using REST APIs and configure an API Gateway (NGINX/Spring Cloud Gateway). Test services using Postman.
9. Create Docker images for microservices, manage Docker containers, and use Docker Hub for image repositories.
10. Deploy a multi-container application using Docker Compose with application and database containers.
11. Install Minikube/Kind and deploy containerized microservices on a Kubernetes cluster using Pods, Deployments, and Services.
12. Configure Kubernetes features including ConfigMaps, Secrets, Service Exposure, Scaling, and Rolling Updates.
13. Apply cloud-native principles by implementing selected 12-Factor App practices such as environment configuration, stateless services, centralized logging, and externalized configuration.
14. Implement observability for microservices using Prometheus and Grafana, monitor application metrics, visualize dashboards, and perform an end-to-end deployment of a cloud-native microservices application.

References:

1. Kubernetes: Up and Running, Brendan Burns, Joe Beda, and Kelsey Hightower, **3rd Edition**, O'Reilly Media, **2022**.
2. Building Microservices, Sam Newman, O'Reilly Media, **2021**.
3. Docker Deep Dive, Nigel Poulton, Leanpub, **2023**.
4. Cloud Native DevOps with Kubernetes, John Arundel and Justin Domingus, O'Reilly Media, **2022**.
5. Software Architecture: The Hard Parts, Neal Ford, Mark Richards, Pramod Sadalage, and Zhamak Dehghani, O'Reilly Media, **2021**.
6. Fundamentals of Software Architecture, Mark Richards and Neal Ford, O'Reilly Media, **2024**.
7. Designing Distributed Systems, Brendan Burns, O'Reilly Media.

Course Outcomes (COs):

1. Explain the fundamentals of cloud computing, cloud infrastructure, virtualization, and microservices architecture, and apply cloud services to develop scalable applications. (PO-1, 2, 3, 5)
2. Design, develop, and deploy cloud-native applications using Docker, Kubernetes, RESTful microservices, and cloud platforms following industry-standard architectural practices. (PO-1, 2, 3, 5)
3. Evaluate cloud-native applications using observability tools, the 12-Factor App methodology, and cloud-native best practices to ensure scalability, reliability, security, and maintainability. (PO-1, 2, 3, 5)

SYSTEM PROGRAMMING LABORATORY	
Course Code: MCSL28	**Credits: 0:0:1**
Pre-requisites: Operating System, Linux CLI and C-programming	**Contact Hours: 28**
Course Coordinator: Dr. Shilpa Chaudhari	

Course Content

This laboratory course helps students to visualize the operating system concepts through various experiments. Following are the list of experiments on the specific operating system concepts.

1. Demonstrate how to avoid the "zombie" state by properly waiting for child processes to exit in a parent process.
2. Configure the Linux kernel according to specific hardware and software requirements.
3. Kernel logging and printk -Create and load an LKM that accepts parameters into the Linux kernel, and observe how parameter values affect the LKM's behavior.
4. Key internal aspects regarding task structure-
   I. Generates a /proc file containing the PIDs and names of all running processes
   II. Create an LKM that changes the priority of a specific process identified by its PID.
   III. Create an LKM that that will display a list of only those tasks which are ‘kernel threads’? (i.e., task->mm == 0). How many ‘kernel threads’ on your list?
5. Key internals regarding memory management within the kernel - examine kernel segment
   I. Create an LKM that prints information about a specific process, including its PID, resident set size (RSS), virtual memory size (VSZ), and command name.
   II. Create an LKM using kmalloc for allocating memory for the struct process_info objects that initializes, tracks, and prints information about running processes.
   III. Create an LKM that allocates memory using both kmalloc and vmalloc, and then compare their characteristics.
6. Load the LKM into the Linux kernel, observe two threads running concurrently and using a mutex for synchronization.
7. Study Linux Namespaces and Process Isolation using Docker

8. Process Isolation and Resource Management using Kubernetes Pods
9. Mini project - demonstrates how system programming concepts (processes, fork, threads) relate to Docker containers and Kubernetes orchestration.

Reference Books:

I. Billimoria, Kaiwan N., Linux Kernel Programming: A comprehensive guide to kernel internals, writing kernel modules, and kernel synchronization. Packt Publishing Ltd, 2021.

II. Bharadwaj, Raghu. Mastering Linux Kernel Development: A kerneldeveloper's reference manual. Packt Publishing Ltd, 2017.

III. Farhan Hasin Chowdhury, The Docker Handbook – Learn Docker for Beginners, https://www.freecodecamp.org/news/the-docker-handbook/, freeCodeCamp press, 2021,

Course Outcomes (COs):

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

1. Apply Linux system programming concepts to create and manage processes, kernel modules, process scheduling, memory management, and synchronization using Linux kernel interfaces (PO1,3,4)
2. Analyze Linux kernel internals by developing Loadable Kernel Modules (LKMs) to examine process structures, memory allocation, kernel threads, and the /proc file system. (PO1,3,4)
3. emonstrate process isolation and resource management using Linux namespaces, Docker containers, and Kubernetes pods, and relate these mechanisms to fundamental system programming concepts(PO1,3,4)

PROFESSIONAL ELECTIVE COURSES-III

ADVANCED SOFTWARE ENGINEERING AND DevOps	
Course Code: MCSE241	**Credits: 3:0:0**
Pre-requisites: Software Engineering	**Contact Hours: 42**
Course Coordinator: Dr. Shilpa Chaudhari	

Course Contents

Unit I

Process Models: A Generic Process Model, Process Assessment and Improvement, Prescriptive Process Models, Specialized Process Models, The Unified Process, Personal and Team Process Models, Process Technology, Product and Process. **Agile Development:** What Is Agility? Agility and the Cost of Change, What Is an Agile Process?, Extreme Programming (XP), Other Agile Process Models, Agile Unified Process (AUP), A Tool Set for the Agile Process.

Unit II

DevOps Culture and Practices: Getting started with DevOps, Implementing CI/CD and continuous deployment. **Continuous Integration and Continuous Delivery**: The CI/CD principles, Using a package manager, Using Jenkins, Using Azure Pipelines, Using GitLab CI. **Requirements Engineering**: Functional and non-functional requirements, The software requirements document, Requirements specification, Requirements engineering processes, Requirements elicitation and analysis, Requirements validation, Requirements management

Unit III

Design Concepts: Design within the Context of Software Engineering, The Design Process, Design Concepts, The Design Model. **Architectural Design:** Software Architecture, Architectural Genres, Architectural Styles, Architectural Design, Assessing Alternative Architectural Designs, Architectural Mapping Using Data Flow U**ser Interface Design:** The Golden Rules, User Interface Analysis and Design, Interface Analysis, Interface Design Steps, WebApp Interface Design

Unit IV

Quality Concepts: What Is Quality?, Software Quality, The Software Quality Dilemma, Achieving Software Quality. **Review Techniques**: Cost Impact of Software Defects, Defect Amplification and Removal, Review Metrics and Their Use, Reviews: A Formality Spectrum, Informal Reviews, Formal Technical Reviews. **Software Quality Assurance:** Background Issues, Elements of Software Quality Assurance, SQA Tasks, Goals, and Metrics, Formal Approaches to SQA, Statistical Software Quality Assurance, Six Sigma for Software Engineering, Software Reliability, The ISO 9000 Quality Standards, The SQA Plan

Unit V

Software Reuse: The reuse landscape, Application frameworks, Software product

lines, COTS product reuse. **Component-based Software Engineering:** Components and component models, CBSE processes, Component composition. **Service-Oriented Architecture:** Services as reusable components, Service engineering, Software development with services. **Aspect-Oriented Software Engineering:** The separation of concerns, Aspects, join points and pointcuts, Software engineering with aspects.

Text Books:

1. Pressman, Roger S. Software Engineering: A Practitioner's Approach, 7th Edition, McGraw-Hill.ISBN 9780073375977, 2010.
2. Krief, Mikael. Learning DevOps, 1st Edition, Packt Publishing Ltd. ISBN 9781838642730, 2019.
3. Sommerville, Ian.Software Engineering, 9th Edition, Pearson Education. ISBN 9780137035151, 2011.

Reference:

1. Verona, Joakim, Practical DevOps, 1st Edition, Packt Publishing Ltd. ISBN 9781785882876,2016.
2. Blaha, Michael, and James Rumbaugh. Object-Oriented Modeling and Design with UML, 2nd Edition, Pearson Education, ISBN 9780130159205, 2004.

Course Outcomes (COs):

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

1. Identify various software development processes and methodologies. (PO: 1, 2, 3 & 5)
2. Apply the DevOps pipeline process starting with continuous integration and continuous deployment principles. (PO: 1, 2, 3 & 5)
3. Recognize the different Design principles, Architectural Design and User Interface Design for developing a high-quality system or product. (PO: 1, 2, 3 & 5)
4. Recognize the need for applying the Software Quality Assurance practice throughout the software process. (PO: 1, 2, 3 & 5)
5. Summarize the need for reusing the software components. (PO: 1, 2, 3 & 5)

CYBER PHYSICAL SYSTEM AND SMART NETWORK	
Course Code: MCSE242	**Credits: 3:0:0**
Pre-requisites: Nil	**Contact Hours: 42**
Course Coordinator: Dr. Sangeetha V.	

Course Contents

UNIT I

Key Features of Cyber-Physical Systems, Synchronous Model – Synchronous Model, Properties of Components, Composing Components, Synchronous Designs. Scheduling Concepts

UNIT II

Safety Requirements - Safety Specifications, Verifying Invariants, Symbolic Search Liveness Requirements - Temporal Logic, Model Checking

UNIT III

Asynchronous Model - Asynchronous Processes, Asynchronous Design Primitives, Asynchronous Coordination Protocols

UNIT IV

Dynamical Systems - Continuous-Time Models, Linear Systems, Designing Controllers - Open-Loop vs. Feedback Controller.
Timed Model - Timed Processes, Timing-Based Protocols, Timed Automata

UNIT V

CPS Applications and Emerging Trends - Smart grid CPS, Intelligent transportation systems, Industrial CPS (Industry 4.0), CPS and IoT convergence, AI-enabled CPS, Research challenges and future directions

Reference Books:

1. “Principles of Cyber-Physical Systems” - Rajeev Alur, MIT Press, 2023
2. "Introduction to Embedded Systems — A Cyber— Physical Systems Approach" - E. A. Lee, Sanjit Seshia", 2017
3. Jeff C. Jensen, Edward A. Lee, and Sanjit A. Seshia, An Introductory Lab in Embedded and Cyber-Physical Systems, http://LeeSeshia.org/lab, First Edition v1.70, 2015.

Course Outcomes (COs):

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

1. Understand the fundamental concepts, architectures, synchronous models, and scheduling mechanisms of Cyber-Physical Systems (CPS).
2. Analyze safety and liveness requirements using formal verification techniques, temporal logic, and model checking methods for reliable CPS design

3. Apply asynchronous process models, coordination protocols, and communication mechanisms in the design of distributed and networked CPS environments.
4. Model dynamical and timed systems, and design feedback-based controllers for real-time CPS applications.
5. Investigate emerging CPS applications such as smart grids, intelligent transportation systems, Industry 4.0, IoT-integrated CPS, and AI-enabled smart infrastructures.

BLOCK CHAIN ESSENTIALS AND DAPPS	
Course Code: MCSE243	**Credits: 3:0:0**
Pre-requisites: Nil	**Contact Hours: 42**
Course Coordinator: Dr. Parkavi A	

Course Contents

Unit I

Introduction, Purpose and Scope, Results of the Public Comment Period, Document Structure, Blockchain Categorization, Permissionless, Permissioned, Blockchain Components, Cryptographic Nonce, Transactions, Asymmetric-Key Cryptography, Ledgers, Blocks, Chaining Blocks, Consensus Models, Forking, Smart Contracts, Blockchain Limitations and Misconceptions, Application Considerations, Additional Blockchain Considerations.

Unit II

Introduction to Cryptography & Cryptocurrencies, Cryptographic Hash Functions, Hash Pointers and Data Structures, Digital Signatures, Public Keys as Identities, A Simple Cryptocurrency, How Bitcoin Achieves Decentralization, Centralization vs. Decentralization, Distributed consensus, Consensus without identity using a block chain, Incentives and proof of work, Mechanics of Bitcoin, Bitcoin transactions, Bitcoin Scripts, Applications of Bitcoin scripts, Bitcoin blocks, The Bitcoin network, Limitations and improvements.

Unit III

Blockchain 3.0: Justice Applications Beyond Currency, Economics, and Markets, Blockchain Technology Is a New and Highly Exective Model for Organizing Activity, Distributed Censorship-Resistant Organizational Models, Namecoin, Digital Art: Blockchain Attestation Services (Notary, Intellectual Property Protection), Blockchain Government, Efficiency and Coordination Applications Beyond Currency, Economics, and Markets, Blockchain Science: Gridcoin, Foldingcoin, Blockchain Genomics, Blockchain Health, Blockchain Learning: Bitcoin MOOCs and Smart Contract Literacy, Blockchain Academic Publishing: Journalcoin, Centralization-Decentralization Tension and Equilibrium, Advanced Concepts.

Unit IV

Ethereum, DApp, Components, EVM, Etherscripter, Hyperledger, Digital Tokens, OmiseGO, EOS, Tether, MetaMask, Wallet Seed, MetaMask Transactions, Objectives of the Hyperledger Project, Mist, Mist wallet, Truffle, Features, Development-Truffle boxes, Truffle Box, Creating a Truffle Box, Community truffle box, Embark, Solidity, Smart Contracts, Statically typed Language, Contract and Interfaces, Hyperledger

Fabric, Mode of operation, Hyperledger Iroha, Components.

Unit V

Hyperledger Sawtooth, Components, Validator registry, Consensus, DApps, Seafood supply chain traceability, Marketplace Digital Asset Exchange, Cello: Features, operator dashboard, Comparison of Bitcoin, Ethereum and Hyperledger, Multichain, Language support, Security, Mining, HydraChain: Smart contracts and HydraChain, IOTA, Corda, Elements Project, deployed Elements., Chain Core, operations available, Development & Security, CoCo Framework, Specialties, Benefits, Tierion, Chainpoint, Benefits of Tierion, BigchainDB, Models, Transaction Models, Block Models.

Text Books:

1. Dylan Yaga, Peter Mell, Nik Roby, Karen Scarfone, Blockchain Technology Overview, NIST Report 8202,US Department of Commerce, Oct 2018.
2. Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, Steven Goldfeder: Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction, Princeton University Press, 2016

Reference Books:

1. Melanie Swan, Block chain, Blueprint for A New Economy, O'Reilly, 2015.
2. Sainul Abideen, Block chain Expert, BlockChain E-Book, Cybrosys Technologies.
   https://www.blockchainexpert.uk/book/blockchain-book.pdf

Course Outcomes (COs):

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

1. Analyze the fundamental elements of block chain technology(PO1,3,4)
2. Demonstrate the application of hashing and public key cryptography in protecting the blockchain (PO1,3,4)
3. Analyse the block chain applications in a structured manner. (PO1,3,4)
4. Develop block chain based solutions and write smart contract using Ethereum Framework. (PO1,3,4)
5. Use smart contract in real world applications (PO 1,3,4,5)

PROFESSIONAL ELECTIVE COURSES-IV

WIRELESS AND MOBILE NETWORKS	
Course Code: MCSE251	**Credits: 3:0:0**
Pre-requisites: Nil	**Contact Hours: 42**
Course Coordinator: Dr. Shilpa S Chaudhari	

Course Contents

Unit I

Introduction to Wireless & Mobile Networks - Evolution of wireless communication systems, Characteristics of wireless channels. Design challenges in wireless and mobile networks.

Wireless Transmission and Multiple Access - Radio propagation models, Noise, interference, and fading, Multiple access techniques: FDMA, TDMA, CDMA, OFDMA, Spread spectrum techniques

Wireless LANs and PANs - IEEE 802.11 architecture and standards, MAC protocols and CSMA/CA, IEEE 802.15 (Bluetooth, ZigBee)

Multiple Antenna Systems - MIMO system models, Spatial diversity and multiplexing, Beamforming techniques, Massive MIMO in 5G, Channel capacity of MIMO systems

Unit II

5G Fundamentals – use cases, network architectures, base station architecture

Air Interface – numerology, radio frames and slots, resource blocks and bandwidth parts, channel bandwidths

Downlink signals and channels – downlink channel mappings, synchronization signals, physical broadcast channel, SS/PBCH blocks and bursts

Unit III

Downlink transmission schemes – PBCH, PDCCH, PDSCH

Flow of Downlink Data – SDAP layer, PDCP layer, RLC layer, MAC layer

System information – master information block, system information block 1

Unit IV

EPC for 5G - Introduction, Key EPC functions, (Enhanced) Dedicated Core Networks ((e) DECOR), Control and User Plane Separation (CUPS

Session management - PDU Session concepts, PDU Session types, User plane handling, Mechanisms to provide efficient user plane connectivity, Session authentication and authorization

Protocols - Introduction ,5G non-access stratum (5G NAS) , NG application protocol (NGAP) , Hypertext transfer protocol (HTTP) , Transport layer security (TLS) , Packet forwarding control protocol (PFCP) , GPRS tunnelling protocol for the User Plane (GTP-U) , Extensible Authentication Protocol (EAP) , IP security (IPSec), Stream

Control Transmission Protocol (SCTP), Generic routing encapsulation (GRE)

Unit V

Quality of Service in Wireless Networks- QoS requirements for wireless and mobile traffic, Scheduling and resource allocation, Cross-layer design approaches

Security in Wireless & Mobile Networks -,Security challenges in wireless environments, Authentication and encryption, Security in WLANs, cellular, and ad hoc networks, Attacks and countermeasures

Emerging Trends and Applications - Vehicular Ad Hoc Networks (VANETs), Device-to-Device (D2D) communication, Overview of beyond beyond 5G (6G)

Reference Books:

1. Chris Johnson, 5G New Radio in Bullets, 1st edition, 2019
2. Stefan Rommer, Peter Hedman, Magnus Olsson, Lars Frid, Shabnam Sultana, Catherine Mulligan, 5G Core Networks - Powering Digitalization, Academic Press, 2020
3. Christopher Cox, An Introduction To LTE, LTE, LTE-Advanced, SAE, VoLTE AND 4G Mobile Communications, Second Edition, John Wiley & Sons, 2014
4. Dohler, M., & Nakamura, T. (2016). *5G mobile and wireless communications technology*. Cambridge University Press.

Course Outcomes (COs):

At the end of the course, a student should be able to:

1. Analyze wireless channel characteristics, multiple access techniques, WLAN/PAN technologies, and MIMO communication systems.. (PO1, 3, 4)
2. Explain the architecture and air interface concepts of LTE and 5G wireless communication systems. (PO1, 2, 3, 4)
3. Analyze downlink transmission procedures, protocol stack operations, and system information flow in 5G networks. (PO1, 2, 3, 4)
4. Evaluate EPC/5G core functions, session management procedures, and communication protocols used in mobile core networks. (PO1, 2, 3, 4, 5)
5. Analyze QoS, security mechanisms, and emerging trends in advanced wireless and mobile networks. (PO1, 2, 3, 4, 5)

GAME THEORY AND MECHANISM DESIGN FOR COMPUTING SYSTEMS	
Course Code: MCSE252	**Credits: 3:0:0**
Pre-requisites: Nil	**Contact Hours: 42**
Course Coordinator: Dr. Rajarajeswari S	

Course Content

Unit I

Basic Mathematical Preliminaries, Key Notions in Game theory, Extensive formgames, Strategic Form games. (TB1: Chapter 33,1,2,3,4)

Unit II

Dominant strategy equilibria, pure strategy Equilibria, mixed strategy and Mixed Strategy Nash Equilibrium, Matrix Games(TB1: Chapter 5,6,7,9)

Unit III

Bayesian games, Introduction to mechanism design , Implementation of social choice functions by mechanisms, Incentive compatibility and revelation theorem (TB1: Chapter 13,14,15,16)

Unit IV

Auctions, Correlates strategies and correlated equilibrium, two persons bargaining problem, coalitional games with transferable utility (TB1: Chapter 20,25,26,27)

Unit V

The core of Coalitional Games, The shapely value, concepts in cooperative gametheory, Stable matching. (TB1: Chapter 28,29,30,31)

Text Book:

1. Y Narahari, "Game Theory and Mechanism Design", IISC Press, World Scientific**.**

Reference Books:

1. Bernhard von Stengel, "Game Theory Basics", Cambridge University Press, 2021
2. M. Maschler, E. Solan and S. Zamir: Game Theory, CUP, 2nd Edition, 2020.
3. Giacomo Bonanno, Game Theory, 2nd Edition 2018.

4. Ken Binmore, Playing for Real: A Text on Game Theory, CUP, 2007.
5. Elliot Mendelson, Introducing Game Theory and Its Applications, CRC 2004.
6. J. Osborne and A. Rubinstein, A Course in Game Theory, MIT press, 1994

Course Outcomes (COs):

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

1. Explain the fundamental concepts of non-cooperative game theory (PO – 1, 2, 3, 4, 5)
2. Apply solution concepts to examples of games, and to state and explain them precisely (PO – 1, 2, 3, 4, 5)
3. Solve unseen games and games with incomplete information that are variants of known examples. (PO – 1, 2, 3, 4, 5)
4. Illustrarte a compelling application of mechanism design. (PO – 1, 2, 3, 4, 5)
5. Develop solution for problems related to cooperative game theory. (PO – 1, 2, 3, 4, 5)

QUANTUM COMPUTING FOR COMPUTER SCIENTISTS	
Course Code: MCSE253	**Credits: 3:0:0**
Pre-requisites: Linear Algebra, Python Programming	**Contact Hours: 42**
Course Coordinator: Dr. Rajarajeswari S	

Course Content

Unit I

Fundamental concepts: Introduction and overview, Introduction to quantum mechanics: Linear algebra, the postulates of quantum mechanics, Quantum Computing software: Introduction to Qiskit, Quantum Qudit simulator, QCAD, Programming a quantum computer: The IBMQ, coding a quantum computer using a simulator to carry out basic quantum measurement and state analysis.

Unit II

Quantum correlations: Bell inequalities and entanglement, Schmidt decomposition, superdense coding, teleportation, density operator, Quantum Computing software: Quack, qasm2circ. Implementation using Qiskit

Unit III

Quantum computation: Quantum Ciruits, Controlled Operations, Measurement, universal Quantum gates, summary, simulation. Quantum Fourier transform and its applications: Quantum fourier transform, phase estimation, Applications. Implementation using Qiskit

Unit IV

Quantum Search Algorithms: QSA, Solovay-Kitaev theorem, Deutsch-Jozsa algorithm, Bernstein-Vazirani Algorithm,Simon's algorithm , Shor's algorithm. Grover's algorithm, Quantum counting, Quantum walk search algorithm
Implementation using Qiskit.

Unit V

Quantum Information: Quantum noise and Quantum operations, Distance measures, Quantum Error Correction, Quantum cryptography: Private Key Cryptography, Privacy amplification and information reconciliation, QKD, privacy and coherentinformation, security of Quantum Key distribution. Implementation using Qiskit.

Text books:

1. M. A. Nielsen &I.Chuang, Quantum Computation and Quantum Information, Cambridge University Press (2016).
2. Eleanor G. Rieffel, and Wolfgang H. Polak, Quantum Computing: A Gentle Introduction, MIT Press, 2014.
3. https://qiskit.org/documentation/

References:

1. Chris Bernhardt, Quantum Computing for Everyone, The MIT Press, Cambridge, 2020
2. David McMahon, Quantum Computing Explained, Wiley-Interscience, IEEE Computer Society, 2008.
3. Phillip Kaye, Raymond Laflamme et. al., An introduction to Quantum Computing, Oxford University Press, 2007.
4. Martin Laforest, The Mathematics of Quantum Mechanics, University of Waterloo, Quantum Cryptography School for young students.

Course Outcomes (COs):

1. Analyze the behavior of basic quantum computation and Simulate basic quantum measurement and state analysis using Qiskit.(PO1, 2, 3, 5)
2. Elaborate on quantum non-locality and simulation of the density operators. (PO-1, 3, 5)
3. Prove basic facts about quantum information channels and Implement information channels in the quantum circuit model. (PO1, 3, 5)
4. Compare, in terms of time complexity, the quantum advantage expected from the quantum algorithms with respect to their classical counterparts. (PO1, 2, 3, 5)
5. Simulate a simple quantum error-correcting code.(PO1, 2, 3, 5)

PROFESSIONAL ELECTIVE COURSES-IV

NATURAL LANGUAGE PROCESSING	
Course Code: MCSE261	**Credits: 3:0:0**
Pre-requisites: Compiler Design	**Contact Hours: 42**
Course Coordinator: Chandrika Prasad	

Course Content

Unit I

Introduction: Knowledge in Speech and Language Processing, Ambiguity, Models and Algorithms; Language, Thought, and Understanding; The State of the Art and The Near-Term Future; Regular Expressions and Automata; Morphology and Finite- State Transducers: Lexicon-free FSTs: The Porter Stemmer, Human Morphological Processing**.**

Unit II

N-grams: Counting Words in Corpora, Smoothing, N-grams for Spelling and Pronunciation, Entropy; Word Classes and Part-of-Speech Tagging: Part- of-Speech Tagging, Rule-based Part-of-speech Tagging, Stochastic Part-of- speech Tagging, Transformation-Based Tagging; Context-Free Grammars for English: Constituency, Context-Free Rules and Trees, Sentence-Level Constructions, The Noun Phrase.

Unit III

Parsing with Context-Free Grammars: The Earley Algorithm; Features and Unification: Feature Structures, Unification of Feature Structures, Features Structures in the Grammar, Implementing Unification, Parsing with Unification Constraints; Lexicalized and Probabilistic Parsing: Probabilistic Context- Free Grammars, Problems with PCFGs.

Unit IV

Representing Meaning: First Order Predicate Calculus, Some Linguistically Relevant Concepts, Related Representational Approaches, Alternative Approaches to Meaning; Semantic Analysis: Syntax-Driven Semantic Analysis, Attachments for a Fragment of English; Lexical Semantics: Relations Among Lexemes and Their Senses, WordNet: A Database of Lexical Relations, The Internal Structure of Words.

Unit V

Discourse: Reference Resolution, Text Coherence, Discourse Structure; Generation: Introduction to Language Generation, An Architecture for Generation; Machine Translation: Language Similarities and Differences, The Transfer Metaphor.

Text Book:

1. Daniel Jurafsky and James H Martin, "Speech and Language Processing: an introduction to Natural Language Processing, Computational Linguistics and Speech Recognition", Prentice Hall, 2nd Edition, 2008.

Reference Book:

1. Tanveer Siddiqui, U.S. Tiwary, "Natural Language Processing and Information Retrieval", Oxford University Press, 2008.

Course Outcomes (COs):

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

1. Interpret how speech and language technology relies on formal models to capture knowledge, and language processing deals with subparts of words(morphology). (PO1,3,4)
2. Illustrate the way *N*-gram tool is used for spelling and pronunciation processing, and part-of-speech tagging mechanism using various categories. (PO1,3,4)
3. Describe feature structures and unification operation which is used to combine them, and probabilistic parsing to capture more syntactic information. (PO1,3,4)
4. Outline representations used to bridge the gap from language to commonsense Knowledge (semantic processing), and meanings associated with lexical items. (PO1,3,4)
5. Emphasize problems that NLP systems face, natural language outputs construction from non-linguistic inputs and machine translation framework approaches. (PO1,3,4)

FULL STACK DEVELOPMENT	
Course Code: MCSE262	**Credits: 3:0:0**
Pre-requisites: Nil	**Contact Hours: 42**
Course Coordinator: Dr. Rajarajeswari S & Swetha	

Course Content

Unit I

Foundations for modern Web Development: Introduction, Cascading Styles Sheet: Concept of CS, Creating Style Sheet, CSS Properties, CSS Styling (Background, Text Format, Controlling Fonts), CSS Id and Class, Box Model (Border, Padding, Margin properties), CSS Advanced (Grouping, Dimension, Display, Positioning, Floating, Align, Pseudo class)

Unit II

Front End Development: JavaScript syntax, Types of Data and Variables, Operations and calculations, The Document Object, Using Events.

JavaScript Advanced: Scopes and Closures, understand "this" and prototypes, OO concepts as applied to JS and prototypal inheritance, Understanding the meaning of asynchronous. Event loops, Promises.

React .JS framework, State management with Redux,, SPA design

Unit III

Database Management: CRUD Operations using Node.js: Event Handling - GET & POST implementation, Use Express.js to create a REST API. Use GET, POST. Connect to NoSQL MongoDBDatabase using Node.js, Implementation of CRUD operations.

GenAI in Web Development: Prompt Engineering Techniques**,** comparing all prompts, AI assisted Coding.

Unit IV

Introduction to Server-side JS Framework – Node.js: Introduction - What is Node.js, Architecture, Feature of Node JS, Installation and setup - Creating webservers with HTTP (Request & Response), Understand dependence management: npm and package.json File system APIs.

Unit V

Introduction to Client-side JS Framework – Building blocks of React,

Create-react-app - Create first React app using this CLI, JSX - Understand what it is and how it's required to create components, Simple functional components, CSS - Load CSS and use it via class Name, props - Passing props to components to make them reusable, Event handling, State - Using class components for storing state (legacy), State via hooks - Using use State hook.

Text Books:

1. Generative AI for Full-Stack Development AI Empowered Accelerated Coding Shantanu Baruah Somerset, NJ, USA ISBN-13 (pbk): 979-8-8688-2073-1

ISBN-13 (electronic): 979-8-8688-2074-8
2. Steven A. Gabarro, " Web Application Design and Implementation: Apache 2, PHP5, MySQL, JavaScript, and Linux/UNIX", Wiley- IEEE Computer Society Press 2007.
3. Nate Murray, Felipe Coury, Ari Lerner and Carlos Taborda, "Ng-book, The Complete Book on Angular", Fullstack.IO, 1st edition, 2016.
4. KrasimirTsonev, "Node.js by Example", Packt Publishing Limited, 2015

Reference Links:

1. Web link for Node.js: https://nodejs.org/en/
2. Web link for MongoDB: https://www.mongodb.com/
3. https://reactjs.org/:

Course Outcomes (COs):

At the end of the course, the student will be able to:

1. Design a web Page using HTML tags (PO 1,2,3,5)
2. Create and style basic web page(PO 1,2,3,5)
3. Explore Node.js features(PO 1,2,3,5)
4. Design Front-end web pages and connect to the Back-end Databases. (PO 1,2,3,5)
5. Create component based web pages using React features(PO 1,2,3,5)

INFORMATION RETRIEVAL AND SEARCH SYSTEMS	
Course Code: MCSE263	**Credits: 3:0:0**
Pre-requisites: Nil	**Contact Hours: 42**
Course Coordinator: Dr. S. Rajarajeswari	

Course Content

Unit I

Introduction: Overview, History of IR, Text Operations: Document preprocessing, Document Clustering, Text Compression, Indexing: Inverted files, **Mathematics for IR**: Set Theory, Mathematical Logic, Probability and Linear algebra, **Classic IR Models**: Boolean Model, Vector space model: tf-idf weighing, Probabilistic Model. Language models for IR: The language model and the query likelihood model.

Unit II

Evaluation Measures: Precision, Recall, Alternative Measures, Reference Collections: TREC, Relevance Feedback and Query Expansion, **Text Classification:** The text classification problem, Flat clustering: Clustering in information retrieval, Problem Statement, Hierarchical clustering Hierarchical agglomerative clustering, Single-link and Complete-link clustering.

Unit III

String Matching algorithms: Knuth Morris Pratt and Rabin Karp, stemming algorithm: Porter, Map reduce algorithms: tf- idf calculation and indexing, **Classification:** Naive Bayes algorithm, clustering k-means algorithm. Machine learning Algorithms: Machine-learned scoring, Result ranking by machine learning

Unit IV

Web search basics: Background and history, Web characteristics, Advertising as the economic model, the search user experience, Index size and estimation, Near- duplicates and shingling, **Web Crawling and Indexing**: Overview, Crawling, Distributing Indexes, Connectivity Servers

Unit V

Link analysis: Web as a graph, PageRank, Hubs and Authorities. **Introduction to Semantic Web**: Purpose, Semantic Web Stack, RDF, RDFS, Ontology, Web ontology language (OWL) and ontology tools.

Text Books:

1. Berthier Ribeiro-Neto, Ricardo Baeza-Yates, Modern Information Retrieval, Pearson Education, 2010.
2. C.D. Manning, P. Raghavan, H. Schütze, Introduction to Information Retrieval, Cambridge University Press, 2008.

Reference Books:

1. Clarke, CL.A., Büttcher, Stefan., Cormack, GordonV.:InformationRetrieval: Implementing and Evaluating Search Engines. MIT Press, 2016.
2. Latha K, Experiment and Evaluation in Information Retrieval Models, CRC Press, 2018
3. David A Grossman, Ophir Frieder: Information Retrieval Algorithms and Heuristics, 2e, Springer,2004
4. Alimohammadi, Dariush:, "Mathematics for Classical Information Retrieval" Zea E-Books Collection. 3, 2010. https://digitalcommons.unl.edu/zeabook/3

Course Outcomes(COs):

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

1. Explain the mathematical foundations and different models of classical information retrieval. (PO 1, 3)
2. Assess the evaluation measures for IR and text classification. (PO 1, 2, 3 ,5)
3. Implement different algorithms for string matching and query processing. (PO1,3,4)
4. Appraise the web research basics and indexing. (PO1,3,4)
5. Discuss the concepts of link analysis and semantic web. (PO1,3,4,5)

SEMESTER III

Professional Elective Course VI and VII	
Course Code: MCSE31x and MCSE21x	**Credits: 3:0:0**
Pre-requisites: -	**Contact Hours: 42**
Course Coordinator: Dr. Parkavi	

NPTEL Courses

Course Content

Professional Elective Course VI	
Subject Code	**Subject Name**
MCSE311	Getting Started with Competitive Programming* https://nptel.ac.in/courses/106106231
MCSE312	Social Networks* https://nptel.ac.in/courses/106106169
MCSE313	Responsible & Safe AI Systems* https://nptel.ac.in/courses/106106472

Professional Elective Course VII	
Subject Code	**Subject Name**
MCSE321	Software Project Management* https://nptel.ac.in/courses/106105218
MCSE322	Computer System Optimizations for ML https://nptel.ac.in/courses/106108003
MCSE323	Generative AI for Computer Vision https://nptel.ac.in/courses/106103001

NOTE: * courses have to be completed using online NPETL platform. If course content is not available on NPTEL then recorded lectures to be provided to students. Evaluation for CIE and SEE to be done by faculty in-charge.

INTERNSHIP / INDUSTRIAL TRAINING	
Course Code: MCSI33	**Credits: 0:0:5**
Pre-requisites: Nil	**Contact Hours: 28**
Course Coordinator: Dr. Shilpa Chaudhari	

Course Content

Internship Work-flow

1. Students submit the initial details including broad area of work and choice of guidein a prescribed format
2. The PG Coordinator along with Head of the department finalizes the guide allocation process
3. Students are given an option to change the guide with mutual consent by applyingthrough prescribed form
4. Students submit the Internship details to guide on the day of registration
5. Problem statement is submitted to PG Coordinator within one week of registration.
6. Students update their progress on weekly basis
7. Weekly meeting with guide is also recorded.
8. Guide evaluates the student on a regular basis according to the rubrics defined in theworksheet for total of 100 marks
9. Evaluation is based on following criteria
    - Tools and technology learnt
    - Relevance of the topic chosen to the current market
    - Report writing
    - Demonstration of tools learnt
    - Presentation of the work carried out as part of internship and vivavoce

Course Outcomes (COs):

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

1. Appraise the requirements of the software projects and project management for industry related projects (PO- 1,2, 3, 4, 5)
2. Design, implement and validate the system according to the requirement andproject plan (PO- 1,2, 3, 4)
3. Identify technical tools and technology relevant to industry project development(PO- 1, 3, 4)
4. Exploit effective tools and techniques for technical report writing andpresentation (PO- 1,2, 3, 4, 5)
5. Work in industry driven project team (PO- 1, 3, 4, 5)

PROJECT WORK - I	
Course Code: MCSP34	**Credits: 0:0:6**
Pre-requisites: Nil	**Contact Hours: 28**
Course Coordinator: Dr. Shilpa Chaudhari	

Course Content

Project Work-flow

1. Students submit the broad area of work and choice of guide in a prescribed format.
2. The PG Coordinator along with Head of the department finalizes the guide allocation process.
3. Students are given an option to change the guide with mutual consent by applyingthrough prescribed form.
4. Students submit the project details to guide on the day of registration
5. Problem statement is submitted to PG Coordinator within one week of registration.
6. Students update their progress on weekly basis
7. Weekly meeting with guide is also recorded.
8. Guide evaluates the student on a regular basis according to the rubrics defined in theworksheet for total of 100 marks
9. Evaluation is based on following criteria
    - Identification of problem domain, relevance of the topic chosen to the current market
    - Study of existing system/literature survey - comparison in terms ofvarious functional and performance parameters
    - Identification of research gaps and formulation of problem statement
    - Literature survey paper draft preparation/submission/publication
    - Objective of the proposed work, methodology and planning for achieving the objectives
    - Software requirement specification document preparation
    - Initial draft of High level design and low level design documentpreparation
    - Implementation details -Tools and technology to be used
    - Demonstration, presentation and viva voce of the work done (in twophases - mid-term review and final review)
    - Regularity
    - Project Phase-I report

Course Outcomes (COs):

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

1. Compare and contrast the available literature in the context of the project for formulation of project problem statement (PO- 1,2, 3, 4, 5)
2. Write the research article on the literature study done (PO- 1,2, 3, 4, 5)
3. Formulate the requirements of the chosen projects and project management plan(PO- 1,2, 3, 4, 5)
4. Utilize technical tools and technology relevant to the project development (PO-1, 3, 4)
5. Exploit effective tools and techniques for technical report writing and presentation (PO- 1,2, 3, 4, 5)

PROJECT WORK - II	
Course Code: MCSP41	**Credits: 0:0:18**
Pre-requisites: Project Work-I	**Contact Hours: 28**
Course Coordinator: Dr. Shilpa Chaudhari	

Course Content

Project Work-flow:

1. Students will continue the work done in project -I. (Course Code: MCSI34 / MCNI34)
2. The PG Coordinator, guide along with Head of the department relook into the project titles and objectives based on the finding from the project work -I (CourseCode: MCSI34 / MCNI34).
3. Students are given an option to change problem statement in the same area whereinthey done the literature survey in project -I. (Course Code: MCSI34 / MCNI34) by applying through prescribed form.
4. Students submit the project details to guide on the day of registration
5. Revised problem statement is submitted to PG Coordinator within one week of registration.
6. Students update their progress on weekly basis
7. Weekly meeting with guide is also recorded.
8. Guide evaluates the student on a regular basis according to the rubrics defined in theworksheet for total of 100 marks
9. Evaluation is based on following criteria
    - Revised (if necessary) problem statement, objective of the proposed work and methodology used for to achieve the objectives,
    - High level design and low level design document preparation
    - Implementation details -Tools and technology used
    - Research article draft on the original work done - preparation/submission/publication
    - Demonstration, presentation and viva voce of the work done in two phase - mid-term review an final review)
    - Regularity
    - Project phase-II report

Course Outcomes(COs):

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

1. Design the software system using SRS document created in Proejct work-I (PO- 1, 3, 4)
2. Develop the dsigned system using relavant technical tools and technology (PO- 1, 3, 4)
3. Test the developed system for result analysis (PO- 1,2, 3, 4, 5)
4. Write the research article on the original work done (PO- 1,2, 3, 4, 5)
5. Demonstrate the work done with the help of effective tools and techniques for technical report writing and presentation (PO- 1,2, 3, 4, 5)