DR. BABASAHEB AMBEDKAR MARATHWADA UNIVERSITY, AURANGABAD

NAAC Reaccredited A Grade

FACULTY OF SCIENCE & TECHNOLOGY

2 Years / 1 Year M.Sc. Computer Science Course Structure

For University Department

(Effective from 2023-24)

COURSE STRUCTURE AS PER GUIDELINES OF NEP 2020

Illustrative Credit distribution structure for two/ one year **M.Sc. Computer Science** Programme with Multiple Entry and Exit options for Discipline Specific Course in Computer Science

A) Eligibility for the course:

i) B.Sc. Computer Science (**OR**) B.Sc. Information Technology **(OR)** B. Sc. Computer Application (Science & Technology) **(OR)** B.E/B. Tech. in Computer Science and Engineering/IT. (**OR**) ii) Any Science Graduate with at least one Optional Subject as Computer Science.

B) Credits and Degrees:

i) A candidate who has successfully completed all the core courses, Elective / Specialized courses and, seminars and project prescribed and or optional service courses approved by the University for the program with prescribed CGPA shall be eligible to receive the degree.

ii) One Credit shall mean one teaching period of one hour per week for one semester (of 15 weeks) for theory courses and two practical / laboratory / field / demonstration hours / week for one semester.

iii) Every student will have to complete at least 88 credits to obtain the master's degree of M. Sc. Computer Science (Post graduate degree)

C) Courses Inclusions:

(i) **Core Course / Mandatory Courses / Discipline Specific Core Courses (DSC):** - A core course is a course that a student admitted to M. Sc. Computer Science program must successfully completed to receive the degree. Normally no theory course shall have more than 4 credits.

(ii) **Discipline Specific Elective Course (DSE):** Means Elective course from the basic subject or specialization. The elective course defined for 4 credits and dedicated for choice of specialization that student want to perceive. The horizontal learning path is to be followed by the student for selection of elective course..

(iii) **Skill Courses (SC):** The service courses will be offered as per the structure of the NEP. This course will be conducted to impart special technology skills to the students. This course is defined for 2 Credits and completely engaged in practical form.

(iv) **On Job Training (OJT)**: The student is required to complete 30 hours on job training in the summer of semester 2 examination.

(iv) Each Course shall include lectures / tutorials / laboratory or field work / Seminar / Practical training / Assignments / midterm and term end examinations/ paper / Report writing or review of literature and any other innovative practice etc., to meet effective teaching and learning needs.

Provision of ONLINE Courses and Credit Transfer

The student(s) is encouraged to undertake equivalence courses from SWAYAM / NPTEL / MICROSOFT PRIME or any other platform. Upon successful completion of the such approved course, in order to get the credit transfer to the academic bank of credit as well as exception of equivalent credit course from the list of academic courses offered by the student, student is required to submit the course completion certificate to the university through college.

D) Multiple entry / Multiple Exits

The course is allowing students for multiple exits as per the guidelines of NEP 2020 and as per the provisions made by the University for the Award of degree. However, these guidelines are updated time to time and the governance related to entry and exit procedure are as per the standard operating procedures of defined by the university.

PROGRAM OBJECTIVES (PO) for M.Sc. COMPUTER SCIENCE

Program objectives for an M.Sc. (Master of Science) in Computer Science course typically aim to provide students with a comprehensive understanding of computer science concepts and practical skills. Following are some broad program objectives earmarked by the department:

1. To equip students with an in-depth understanding of advanced topics in computer science, including algorithms, data structures, artificial intelligence, machine learning, internet of things, and more (**Advanced Knowledge**).

2. To foster the ability to critically analyse complex problems in computer science and conduct independent research to propose innovative solutions (**Research and Analysis**).

3. To develop strong programming skills in multiple languages (wide array of programming basket) and paradigms, enabling students to design and implement software systems and applications (**Programming Proficiency**).

4. To enhance problem-solving and logical thinking skills to tackle real-world challenges in computer science and related interdisciplinary domains (**Problem-Solving Abilities**).

5. To cultivate a sense of ethical responsibility and professionalism among students, emphasizing the importance of adhering to legal and ethical standards in computing (**Ethical and Professional Awareness**).

6. To promote effective teamwork, communication, and leadership skills to work collaboratively in diverse, multidisciplinary projects (**Collaboration and Communication**).

7. To keep students abreast of the latest developments and emerging technologies in the field of computer science, such as cloud computing, cybersecurity, internet of things (IoT), and blockchain (**Emerging Technologies**).

8. To provide opportunities for hands-on experience through case studies, projects, internships, and practical assignments, ensuring students can apply theoretical knowledge to real-world scenarios (**Practical Experience**:).

9. To encourage an entrepreneurial mindset and foster innovation, allowing students to develop novel solutions and potentially start their ventures in the technology industry (**Innovation and Entrepreneurship**).

10. To teach students how to critically evaluate existing research, publications, and technological advancements in computer science (**Critical Evaluation**).

11. To in-still the value of continuous learning and adaptability in an ever-evolving field like computer science (**Adaptability and Lifelong Learning**).

PROGRAM SPECIFIC OBJECTIVES (PSO) for M.Sc. COMPUTER SCIENCE

Program Specific Objectives (PSOs) are more focused and concrete statements that describe the specific outcomes expected from a Master of Science (M.Sc.) in Computer Science program. These objectives should align with the broader program objectives mentioned earlier.

PSO 1: Advanced Problem Solving and Analysis Skills: Students of the program will be able to identify, analyse, and solve complex problems in computer science using theoretical knowledge and practical skills gained during the course.

PSO 2: Proficiency in Software Development: Students will demonstrate expertise in designing, implementing, testing, and maintaining software systems and applications using various programming languages and development methodologies.

PSO 3: Research and Innovation Capabilities: Students will develop the ability to conduct independent research, contribute to existing knowledge in computer science, and propose innovative solutions in specialized areas.

PSO 4: Specialization Expertise: Students will acquire in-depth knowledge and expertise in their chosen specialization, such as artificial intelligence, data science, cybersecurity, software engineering, etc.

PSO 5: Effective Communication and Collaboration: Students will cultivate strong communication, teamwork, and leadership skills to work effectively in multidisciplinary teams and convey technical concepts to diverse audiences.

PSO 6: Ethical and Professional Practices: Students will understand the ethical and legal implications of computer science applications and adhere to professional standards in their roles as computer scientists.

PSO 7: Adaptability and Lifelong Learning: Graduates will embrace a growth mindset and exhibit the ability to adapt to emerging technologies and continue their learning beyond the program.

PSO 8: Software Development Lifecycle Proficiency: Students will gain hands-on experience in each phase of the software development lifecycle, including requirements gathering, design, implementation, testing, deployment, and maintenance.

SEMESTER WISE PROGRAM STRUCTURE

Semester I

Course Type	Course Code	Course Name	Teaching Hrs		Credits Assigned		Total Credits
			TH	PR	TH	PR	
Discipline Specific Courses (Major Mandatory) - DSC	UDCS/MJT/500	Introduction to Algorithms: Theory, Practice and Case study	02	-	02	-	14
	UDCS/MJT/501	Data Visualization	02	-	02	-	
	UDCS/MJT/502	Sensors and Actuators	02	-	02	-	
	UDCS/MJP/500	Practical Based on Introduction to Algorithms: Theory, Practice and Case study	-	04	-	02	
	UDCS/MJP/501	Practical Based on Data Visualization	-	04	-	02	
	UDCS/MJP/502	Practical Based on Sensors and Actuators	-	04	-	02	
Skill / Advance Course	UDCS/MJP/503 - 507	Programming-1*	00	04	-	02	
Discipline Specific Electives - DSE	UDCS/DSET/520 – 524	Elective-1#	02	-	02	-	02
	UDCS/DSEP/520 – 524	Elective-1#	-	04	-	02	02
Research Methodology	UDCS /RMT/530	Research Methodology	04	-	04	-	04
		Total	**12**	**20**	**12**	**10**	**22**

Note: *,# Student is advised to select the any one course from the pool of courses, however horizontal selection of courses to be followed at the time of selection of the course .MJT means mandatory theory, MJP means mandatory theory, DSET **Discipline Specific Electives Theory and** DSEP **Discipline Specific Electives Practical**

Semester II

Course Type	Course Code	Course Name	Teaching Hrs.		Credits Assigned		Total Credits
			TH	PR	TH	PR	
Discipline Specific Courses (Major Mandatory) - DSC	UDCS /MJT/550	Digital Image Processing & Analytics	02	-	02	-	14
	UDCS /MJT/551	Statistical Machine Learning	02	-	02	-	
	UDCS /MJT/552	Microcontroller Programming using Python	02	-	02	-	
	UDCS /MJP/550	Practical Based on Digital Image Processing & Analytics	-	04	-	02	
	UDCS /MJP/551	Practical Based on Statistical Machine Learning	-	04	-	02	
	UDCS /MJP/552	Practical Based on Microcontroller Programming using Python	-	04	-	02	
Skill / Advance Course	UDCS /MJP/553 – 557	Programming-2*	00	04	-	02	
Discipline Specific Electives - DSE	UDCS /DSET/570 - 574	Elective-2#	02	-	02	-	02
	UDCS /DSEP/570 - 574	Elective-2#	-	04	-	02	02
On-Job Training	UDCS /OJT/590	On-Job Training / Field Project	-	08	-	04	04
		Total	**08**	**28**	**08**	**14**	**22**

Note: *,# Student is advised to select the any one course from the pool of courses, however horizontal selection of courses to be followed at the time of selection of the course.MJT means mandatory theory, MJP means mandatory theory, DSET **Discipline Specific Electives Theory and** DSEP **Discipline Specific Electives Practical**

Semester III

Course Type	Course Code	Course Name	Teaching Hrs.		Credits Assigned		Total Credits
			TH	PR	TH	PR	
Discipline Specific Courses (Major Mandatory) - DSC	UDCS/MJT/600	Computer Vision	02	-	02	-	14
	UDCS /MJT/601	Data Mining and Warehousing using Python	02	-	02	-	
	UDCS /MJT/602	Internet of Things (IoT)	02	-	02	-	
	UDCS /MJP/600	Practical Based on Computer Vision	-	04	-	02	
	UDCS /MJP/601	Practical Based on Data Mining and Warehousing using Python	-	04	-	02	
	UDCS /MJP/602	Practical Based on Internet of Things (IoT)	-	04	-	02	
Skill / Advance Course	UDCS /MJP/603-607	Programming-3*	00	04	-	02	
Discipline Specific Electives - DSE	UDCS /DSET/620 – 624	Elective-3#	02	-	02	-	04
	UDCS /DSEP/620 – 624	Elective-3#	-	04	-	02	
Research Project	UDCS /RP/649	Research Project -1	00	08	04		04
		Total	**08**	**28**	**12**	**10**	**22**

Note: *,# Student is advised to select the any one course from the pool of courses, however horizontal selection of courses to be followed at the time of selection of the course. MJT means mandatory theory, MJP means mandatory theory, DSET **Discipline Specific Electives Theory and** DSEP **Discipline Specific Electives Practical**

Semester IV

Course Type	Course Code	Course Name	Teaching Hrs		Credits Assigned		Total Credits
			TH	PR	TH	PR	
Discipline Specific Courses (Major Mandatory) - DSC	UDCS/MJT/650	Artificial Intelligence	02	-	02	-	12
	UDCS/MJT/651	Data Analytics	02	-	02	-	
	UDCS/MJT/652	IoT and Cloud : Concept, Case Study and Application	02	-	02	-	
	UDCS/MJP/650	Practical Based on Artificial Intelligence	-	04	-	02	
	UDCS/MJP/651	Practical Based on Data Analytics	-	04	-	02	
	UDCS/MJP/652	Practical Based on IoT and Cloud : Concept, Case Study and Application	-	04	-	02	
Discipline Specific Electives - DSE	UDCS/DSET/670 – 674	Elective-4#	02	-	02	-	04
	UDCS/DSEP/670 – 674	Elective-4#	-	04	-	02	-
Research Project	UDCS /RP/699	Research Project -2	00	12	-	06	06
		Total	**08**	**28**	**08**	**14**	**22**

Note: *,# Student is advised to select the any one course from the pool of courses, however horizontal selection of courses to be followed at the time of selection of the course. MJT means mandatory theory, MJP means mandatory theory, DSET **Discipline Specific Electives Theory and** DSEP **Discipline Specific Electives Practical**

Skill / Advance Course (Programming Elective Group Basket)

Programming Group	**Programming – 1** UDCS/MJP/503-507	**Programming-2** UDCS/MJP/553-557	**Programming -3** UDCS/MJP/603-607
Java Group	Core Java	Advance Java	Android
Microsoft Group	Advance C++	VB.NET	C# NET
Open Group	Python	Advanced Python	Open Web Programming (PHP)
Developer Group	Rust	GO	Kotlin
Web Scripting Group	VB & JavaScript	NodeJS	React

Note: *,# Student is advised to select the any one course from the pool of courses, however horizontal selection of courses to be followed at the time of selection of the course. MJT means mandatory theory, MJP means mandatory theory, DSET **Discipline Specific Electives Theory and** DSEP **Discipline Specific Electives Practical**

Discipline Specific Electives - DSE (Elective Group Basket)

Elective Group	**Elective 1** UDCS/DSET/520-524 & UDCS/DSEP/520-524	**Elective 2** UDCS /DSET/570-574 & UDCS /DSEP/570-574	**Elective 3** UDCS /DSET/620-624 & UDCS /DSEP/620-624	**Elective 4** UDCS /DSET/670-674 & UDCS /DSEP/670-674
Pattern Analysis & Machine Intelligence	Soft Computing	Fuzzy Systems : Theory, Application & Case Study	Video Processing	Pattern Recognition
Remote Sensing and Geospatial Technology	Fundamental of Satellite Remote Sensing	GIS	Remote Sensing And Digital Image Analysis	Hyperspectral Image Analysis
Security	Network Security	Cyber Security	Cyber Forensics: Tools, Techniques and Case Studies	Cryptography & Blockchain
Natural Language Processing	Linguistic Fundamentals: Understanding Language Structure and Analysis	Semantics and Pragmatics	Natural Language Processing	AI Chatbot Services and Applications
Quantum Computing	Introduction to Quantum Computing	Quantum Bits (Qbits)	Quantum Algorithms	Introduction to Advanced Quantum Technology and Applications

Note: *,# Student is advised to select the any one course from the pool of courses, however horizontal selection of courses to be followed at the time of selection of the course. MJT means mandatory theory, MJP means mandatory theory, DSET **Discipline Specific Electives Theory and** DSEP **Discipline Specific Electives Practical**

Semester I

Course Type(**Mandatory)**			
Course Code	**Course Title**	**Credits**	**Contact Hours**
CCS/MJT/500	**Introduction to Algorithms: Theory, Practice, and Case Studies**	2	**2hrs/per week**
CCS/MJP/500	Practical Based on Introduction to Algorithms: Theory, Practice, and Case Studies	2	**4hrs/per week**

Course Description:

This course is designed to provide students with a comprehensive introduction to algorithms, their analysis, and their practical implementation. Students will gain a solid understanding of fundamental algorithmic concepts and techniques, and how they apply to various problem domains. The course will cover both theoretical aspects of algorithm design and practical applications through case studies. Students will also be introduced to relevant reference books and resources for further study.

Prerequisites:

- Basic programming skills in a high-level language (e.g., Python, Java, C++)
- Understanding of fundamental data structures (e.g., arrays, linked lists, stacks, queues)

Course Objectives (CO):

- Understand fundamental concepts and terminologies related to algorithm design and analysis.
- Analyse the efficiency of algorithms and make informed choices based on their performance characteristics.
- Apply various algorithmic techniques to solve a wide range of problems.
- Gain hands-on experience in implementing and evaluating algorithms using programming languages.
- Recognize the relevance of algorithms in real-world applications through case studies.
- Develop critical thinking and problem-solving skills required for algorithm design and analysis.

Learning Outcomes (LO):

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

- Design and implement efficient algorithms to solve computational problems.
- Evaluate the time and space complexity of algorithms using mathematical analysis and empirical studies.
- Select appropriate data structures for specific problem domains.
- Apply algorithmic techniques, such as greedy algorithms and dynamic programming, to tackle complex problems.
- Analyze and compare different algorithms to make informed decisions based on their efficiency and effectiveness.
- Apply algorithms in practical scenarios and understand their impact on various fields of study.

Course Outline:

Unit 1: Introduction to Algorithms and Analysis - Overview of algorithms and their importance, Algorithm analysis: time complexity, space complexity, and big-O notation, Asymptotic analysis: best-case, worst-case, and average-case analysis, Analysis of iterative and recursive algorithms. Fundamental Algorithm Design Techniques - Brute force and exhaustive search, Divide and conquer: merge sort, quicksort, and binary search, Greedy algorithms: knapsack problem, minimum spanning tree, Dynamic programming: Fibonacci sequence, longest common subsequence

Unit 2: Graph Algorithms - Graph representation: adjacency matrix, adjacency list, Breadth-first search (BFS) and depth-first search (DFS), Shortest path algorithms: Dijkstra's algorithm, Bellman-Ford algorithm, Minimum spanning tree algorithms: Prim's algorithm, Kruskal's algorithm. Searching and Sorting Algorithms - Linear search and binary search, Bubble sort, insertion sort, and selection sort, Merge sort and quicksort, Radix sort and counting sort,

Unit 3: Advanced Topics in Algorithms - NP-completeness and computational intractability Approximation algorithms and heuristics, Network flow algorithms: Ford-Fulkerson algorithm, Edmonds-Karp algorithm, String algorithms: pattern matching, string matching, and string compression. Case Studies and Practical Applications - Real-world case studies showcasing the application of algorithms, Algorithmic problem-solving in different domains (e.g., scheduling, network optimization), Implementation and analysis of algorithms using programming languages

Reference Books:

"Introduction to Algorithms" by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein.

"Algorithms" by Robert Sedgewick and Kevin Wayne.

"The Algorithm Design Manual" by Steven S. Skiena.

"Algorithm Design: Foundations, Analysis, and Internet Examples" by Michael T. Goodrich and Roberto Tamassia.

"Algorithms Unlocked" by Thomas H. Cormen.

Course Type(**Mandatory)**			
Course Code	**Course Title**	**Credits**	**Contact Hours**
CCS/MJT/501	**Data Visualization**	2	**2hrs/per week**
CCS/MJP/501	Practical Based on Data Visualization	2	**4hrs/per week**

Course Description:

This course on Data Visualization provides a comprehensive overview of the principles, techniques, and tools used to effectively visualize data. Students will gain a solid foundation in data visualization concepts and develop practical skills to create compelling visual representations of data. The course will cover various visualization types, data analysis techniques, and best practices in designing visualizations. Through case studies and hands-on exercises, students will explore real-world applications of data visualization in different domains.

Prerequisites:

- Basic understanding of statistics and data analysis concepts
- Familiarity with spreadsheet software (e.g., Microsoft Excel, Google Sheets)
- Basic programming knowledge (preferred but not mandatory)

Course Objectives (CO):

- Understand the fundamental principles and techniques of data visualization.
- Gain proficiency in selecting appropriate visualization techniques for different types of data.
- Develop skills in creating visually appealing and effective data visualizations.
- Learn to design interactive and engaging visualizations to facilitate data exploration.
- Apply data visualization concepts to real-world case studies and datasets.
- Enhance storytelling abilities by incorporating data visualization techniques.
- Cultivate ethical awareness and responsibility in data visualization practices.

Learning Outcomes (LO):

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

- Demonstrate a comprehensive understanding of data visualization principles and techniques.
- Evaluate and select appropriate visualization techniques based on data types and objectives.
- Design and create visually compelling and informative data visualizations.
- Develop interactive visualizations to facilitate data exploration and user engagement.
- Apply data visualization skills to analyze and present complex datasets effectively.
- Incorporate storytelling elements into data visualizations to convey insights clearly.

- Recognize and address ethical considerations in data visualization practices.

Course Outline:

Unit 1: Introduction to Data Visualization - Definition and importance of data visualization, Types of data visualization techniques, Overview of data visualization tools and software. Principles of Effective Data Visualization - Gestalt principles and visual perception, Color theory and best practices, Layout and composition guidelines, Data storytelling and narrative techniques

Unit 2: Data Types and Visualization Technique - Visualizing numerical data: line charts, bar charts, scatter plots, etc, visualizing categorical data: pie charts, stacked bar charts, treemaps, etc, Visualizing temporal data: time series plots, heatmaps, calendars, etc, Visualizing spatial data: maps, choropleth maps, cartograms, etc. Interactive Data Visualization - Introduction to interactive visualization tools, Adding interactivity using tooltips, filters, and selection, Creating interactive dashboards and exploratory visualizations, Designing for user engagement and ease of use.

Unit 3: Advanced Data Visualization Techniques - Network visualization and graph-based data, Hierarchical and tree-based visualizations, 3D and multidimensional visualizations, Geospatial and geo-visualization techniques. Data Visualization and Storytelling - Communicating insights through effective data visualization, designing visual narratives and story arcs, integrating text, annotations, and annotations into visualizations, Presenting data visualizations to diverse audiences. Data Visualization Ethics and Best Practices - Ethical considerations in data visualization, Data accuracy, integrity, and representation, Avoiding bias and misleading visualizations, Responsible data storytelling and interpretation.

<u>Case Studies</u>:

1. Visualizing Global Health Data: Analysing and presenting health indicators across countries to identify patterns and disparities.

2. Interactive Sales Dashboard: Designing an interactive dashboard to explore sales data and identify trends, regions, and product performance.

3. Network Analysis: Visualizing social networks or organizational structures to reveal connections and influence patterns.

4. Geospatial Data Visualization: Mapping and analyzing geographic data, such as population density, distribution of resources, or climate patterns.

5. Time Series Visualization: Analyzing temporal data, such as stock prices or weather patterns, to identify trends and make predictions.

Recommended Reference Book

"Data Visualization: A Practical Guide" by Andy Kirk, SAGE Publications Ltd.2021

Course Type(**Mandatory)**			
Course Code	**Course Title**	**Credits**	**Contact Hours**
CCS/MJT/502	**Sensors and Actuators**	2	**2hrs/per week**
CCS/MJP/502	Practical Based on Sensors and Actuators	2	**4hrs/per week**

Course Description:

This course provides a comprehensive introduction to sensors and actuators, their principles of operation, and their applications in various fields. Students will gain an understanding of the fundamental concepts behind sensor and actuator technology, explore different types of sensors and actuators, and learn about their integration into real-world systems. The course will include theoretical concepts, practical examples, case studies, and reference materials to enhance the learning experience.

Prerequisites:

- Basic knowledge of electronics and circuits
- Understanding of fundamental physics principles
- Familiarity with microcontrollers and programming
- Knowledge of measurement and control systems

Course Objectives (CO):

- To provide an understanding of sensor and actuator technologies, their principles of operation, and their role in various systems.
- To explore the design considerations and specifications for selecting appropriate sensors and actuators for different applications.
- To develop knowledge and skills in designing interface circuits and systems for sensors and actuators.
- To understand the integration techniques and challenges involved in incorporating sensors and actuators into larger systems.
- To analyze real-world case studies and applications to gain practical insights into sensor and actuator design.

Learning Outcomes (LO):

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

- Identify different types of sensors and actuators and explain their principles of operation.
- Evaluate and select suitable sensors and actuators based on application requirements.
- Design and implement signal conditioning circuits for sensors and actuators.
- Integrate sensors and actuators into larger systems, considering packaging, communication, and control aspects.
- Analyse and troubleshoot sensor and actuator-related issues in practical applications.
- Apply knowledge gained from case studies to solve design challenges in sensor and actuator applications.

Course Outline:

Unit 1: Introduction to Sensors and Actuators - Overview of sensors and actuators, Importance and applications of sensors and actuators, Sensor and actuator classifications, Principles of operation, Sensor characteristics: accuracy, resolution, sensitivity, range, etc. Sensor Technologies - Resistive sensors: thermistors, strain gauges, potentiometers, Capacitive sensors: proximity sensors, touchscreens, Inductive sensors: proximity sensors, encoders, Optical sensors: photodiodes, phototransistors, optical encoders, Piezoelectric sensors: accelerometers, pressure sensors, Magnetic sensors: Hall effect sensors, magnetometers, Chemical and gas sensors: pH sensors, gas detectors

Unit 2: Actuator Technologies - Electric actuators: motors, solenoids, relays. Hydraulic actuators: pumps, valves, cylinders. Pneumatic actuators: compressors, valves, actuators. Piezoelectric actuators: piezoelectric motors, actuators. Shape memory alloy actuators: shape memory alloys (SMAs) - Electroactive polymer actuators: dielectric elastomers, ferroelectrics. Sensor and Actuator Integration - Signal conditioning and amplification, Sensor and actuator interfacing, Communication protocols: I2C, SPI, UART, Microcontrollers and embedded systems, Introduction to control systems

Unit 3: Applications and Case Studies - Sensors and actuators in robotics and automation, Automotive applications: ABS, airbags, engine control, Healthcare applications: medical devices, prosthetics, Industrial applications: process control, monitoring systems. Smart home applications: home automation, security systems, Environmental sensing and monitoring. Emerging Trends and Future Directions - Advances in sensor and actuator technologies, Internet of Things (IoT) and sensor networks, Wearable sensors and actuators, Artificial intelligence and machine learning in sensor systems, Ethical considerations and challenges in sensor and actuator design

Recommended Reference Books:

"Sensors and Actuators: Engineering System Instrumentation" by Clarence W. de Silva

"Principles of Sensors and Actuators" by Sergio S. C. Jr. and Andrey F. S. Machado

"Smart Sensors and MEMS: Intelligent Devices and Microsystems for Industrial Applications" by Sergey Y. Yurish

"Sensor Technology Handbook" by Jon S. Wilson

"Introduction to Mechatronics and Measurement Systems" by David G. Alciatore and Michael B. Histand

"Sensor Systems: Fundamentals and Applications" by Clarence W. de Silva

Course Type(**Mandatory)**			
Course Code	**Course Title**	**Credits**	**Contact Hours**
CCS/RMT /530	**Research Methodology (Computer Science)**	4	**4hrs/per week**

Course Description:

This course is designed to provide computer science students with a comprehensive understanding of computational and statistical methods used in research methodology. Students will learn various techniques and tools to analyse, interpret, and present research data in computer science. The course will also include case studies to illustrate the application of these methods in real-world scenarios. Students will gain practical skills in designing experiments, collecting and analysing data, and drawing meaningful conclusions. Reference books will be provided to supplement the course materials and enhance the students' knowledge.

Prerequisites:

- Basic understanding of computer science concepts
- Familiarity with programming languages (e.g., Python, R, or MATLAB)
- Knowledge of introductory statistics

Course Objectives (CO):

- To introduce students to the fundamental principles of research methodology in computer science.
- To provide students with a comprehensive understanding of computational and statistical methods used in computer science research.
- To equip students with the skills to collect, pre-process, analyse, and interpret data for research purposes.
- To enable students to apply appropriate statistical techniques for hypothesis testing and inference.
- To familiarize students with machine learning algorithms and their applications in computer science research.
- To develop students' ability to design and analyse experiments in the context of computer science research.
- To enhance students' critical thinking and problem-solving skills through case studies and practical assignments.

Learning Outcomes (LO):

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

- Formulate research questions and select appropriate research methodologies.
- Collect, pre-process, and analyse data using computational and statistical techniques.
- Apply hypothesis testing and statistical inference to draw meaningful conclusions from data.
- Build and evaluate regression and machine learning models for predictive analysis.
- Design and analyse controlled experiments to investigate research hypotheses.
- Demonstrate an understanding of real-world applications of computational and statistical methods in computer science research.

- Critically evaluate research studies and identify strengths and limitations in their methodology.

Course Outline:

Unit 1: Introduction to Research Methodology - Overview of research methodology in computer science, Research design and problem formulation, Literature review and identifying research gaps, Ethical considerations in research

Unit 2: Data Collection and Pre-processing - Data collection techniques: surveys, interviews, observations, etc. Data pre-processing and cleaning, Handling missing data and outliers, Exploratory data analysis, Case study: Visualizing and summarizing real-world datasets

Unit 3: Statistical Analysis - Descriptive statistics: measures of central tendency, dispersion, etc., Hypothesis testing and statistical significance, Parametric and non-parametric tests, Analysis of variance (ANOVA) and regression analysis, Case study: Applying statistical inference techniques to analyse experimental data

Unit 4: Computational Analysis - Introduction to machine learning algorithms, Supervised and unsupervised learning techniques, feature selection and dimensionality reduction, Evaluation metrics for Computational Methods and Techniques, Case study: Building a computational model for classification or clustering

Unit 5: Presenting Research Findings - Effective data visualization techniques, Scientific writing and report preparation, Presenting research findings in conferences and journals, Peer review process and publication ethics.

Assessment Methods:

- Assignments and quizzes to assess understanding of concepts and techniques
- Course project report and presentation evaluation
- Participation in class discussions and group activities

Recommended Reference Books:

"Research Methodology: A Step-by-Step Guide for Beginners" by Ranjit Kumar

"Designing and Conducting Mixed Methods Research" by John W. Creswell and Vicki L. Plano Clark

"Statistical Methods for Computer Science" by Walter D. Wallis

"Pattern Recognition and Machine Learning" by Christopher M. Bishop

"Data Science for Business" by Foster Provost and Tom Fawcett

"The Elements of Statistical Learning: Data Mining, Inference, and Prediction" by Trevor Hastie, Robert Tibshirani, and Jerome Friedman

"Visualization Analysis and Design" by Tamara Munzner

Semester – II

Course Type(**Mandatory)**			
Course Code	**Course Title**	**Credits**	**Contact Hours**
CCS/MJT/550	**Digital Image Processing & Analytics**	2	**2hrs/per week**
CCS/MJP/550	Practical Based on Digital Image Processing & Analytics	2	**4hrs/per week**

Course Description:

This credit introductory course on Digital Image Processing & Analytics provides students with a solid foundation in the fundamentals of image processing techniques and the application of analytics methods to extract meaningful insights from images. The course covers both theoretical concepts and practical skills necessary for processing and analyzing digital images. Through case studies and hands-on exercises, students will develop a comprehensive understanding of image processing algorithms, tools, and their real-world applications.

Prerequisites:

- Basic knowledge of mathematics, including linear algebra and calculus.
- Understanding of programming concepts and proficiency in a programming language (e.g., Python, MATLAB).
- Familiarity with basic image processing concepts and terminology.

Course Objectives (CO):

- To provide students with a solid foundation in digital image processing techniques.
- To introduce students to various image enhancement, restoration, and segmentation methods.
- To familiarize students with feature extraction and representation techniques for image analysis.
- To teach students about image compression and coding algorithms.
- To enable students to understand and implement object detection and recognition algorithms.
- To provide students with knowledge and skills in image registration, fusion, and interpretation.
- To expose students to real-world case studies and applications of digital image processing.

Learning Outcomes (LO):

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

- Understand the fundamental concepts and principles of digital image processing.
- Apply various image enhancement techniques to improve image quality.
- Restore and reconstruct images by removing noise and blurring artifacts.
- Segment images into meaningful regions using different techniques.
- Extract relevant features from images for further analysis and interpretation.
- Compress and code images using different algorithms to achieve efficient storage and transmission.
- Detect and recognize objects in images using different algorithms and machine learning approaches.
- Register and fuse images to create composite images with enhanced information.
- Analyse and interpret images for various applications, such as medical imaging, remote sensing, and computer vision.
- Analyse and solve real-world problems through case studies and practical assignments.

Course Outline:

Unit 1: Introduction to Digital Image Processing - Overview of digital image processing, Image acquisition and representation, Image enhancement techniques, Image compression and storage. Image Filtering and Restoration - Spatial domain filtering, Frequency domain filtering, Image denoising and restoration. Case study: Image restoration in medical imaging

Unit 2: Image Segmentation - Thresholding techniques, Region-based segmentation, Edge detection and boundary extraction Case study: Object detection in computer vision. Feature Extraction and Representation - Feature extraction methods, Texture analysis, Shape analysis and recognition, Case study: Facial recognition and biometrics. Image Analysis and Classification - Image classification techniques, Statistical pattern recognition, Machine learning for image analysis, Case study: Image-based disease diagnosis

Unit 3: Image Registration and Fusion - Image registration techniques, Multi-sensor image fusion, Applications of image fusion, Case study: Remote sensing and GIS. Introduction to Deep Learning for Image Processing - Basics of deep learning, Convolutional neural networks (CNNs), Transfer learning in image analysis, Case study: Image recognition with CNNs. Ethical Considerations and Challenges in Image Analytics - Privacy and security concerns, Bias and fairness in image analysis, Ethical guidelines and best practices, Case study: Ethical issues in facial recognition

Case Studies:

1. Medical Imaging: Enhancing and analysing medical images for diagnosis and treatment planning.

2. Computer Vision: Object detection and recognition in real-world images and videos.

3. Remote Sensing and GIS: Analysing and integrating satellite images for environmental monitoring and urban planning.

4. Biometrics: Facial recognition and identification systems for security applications.

5. Image-based Disease Diagnosis: Using machine learning for early detection of diseases through medical imaging.

6. Medical Image Analysis: Analysing medical images (e.g., X-rays, MRIs) for diagnosis and treatment planning.

7. Satellite Image Processing: Processing satellite imagery for land cover classification and change detection.

8. Face Recognition: Implementing face recognition algorithms for security and biometrics applications.

9. Autonomous Driving: Applying image processing techniques for object detection and lane detection in autonomous vehicles.

Recommended Reference Books:

"Digital Image Processing" by Rafael C. Gonzalez and Richard E. Woods

"Digital Image Processing and Analysis: Human and Computer Vision Applications with CVIPtools" by Scott E. Umbaugh

"Computer Vision: Algorithms and Applications" by Richard Szeliski

"Pattern Recognition and Machine Learning" by Christopher M. Bishop

"Deep Learning" by Ian Goodfellow, Yoshua Bengio, and Aaron Courville

"Remote Sensing and Image Interpretation" by Thomas Lillesand, Ralph W. Kiefer, and Jonathan W. Chipman

"Biometrics: Identity Verification in a Networked World" by Anil K. Jain, Arun A. Ross, and Karthik Nandakumar

Course Type(**Mandatory)**			
Course Code	**Course Title**	**Credits**	**Contact Hours**
CCS/MJT/551	**Statistical Machine Learning**	2	**2hrs/per week**
CCS/MJP/551	Practical Based on Statistical Machine Learning	2	**4hrs/per week**

Course Description:

This introductory credit course aims to provide students with a solid foundation in statistical machine learning techniques. The course covers essential concepts and algorithms used in statistical machine learning, with a focus on their practical applications through case studies. Students will gain a comprehensive understanding of the underlying principles and tools used in this field, enabling them to analyse and interpret data, make predictions, and solve real-world problems. The course will also introduce students to relevant reference books to further explore the topics covered.

Prerequisites:

- Basic knowledge of probability and statistics
- Familiarity with programming (Python preferred)
- Understanding of linear algebra concepts
- Some exposure to calculus

Course Objectives (CO):

- To provide students with a comprehensive understanding of statistical machine learning techniques and algorithms.
- To equip students with the knowledge and skills necessary to apply statistical machine learning to real-world problems.
- To develop critical thinking and analytical skills for model evaluation, selection, and optimization.
- To introduce students to popular machine learning frameworks and libraries for implementation.

Learning Outcomes (LO):

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

- Understand the fundamental concepts and principles of statistical machine learning.
- Apply supervised and unsupervised learning algorithms to analyze and interpret data.
- Evaluate and compare the performance of different machine learning models.
- Implement machine learning algorithms using programming languages and frameworks.
- Apply statistical machine learning techniques to solve real-world problems in various domains.
- Demonstrate the ability to interpret and present the results of machine learning models effectively.

Course Outline:

Unit 1: Introduction to Statistical Machine Learning - Overview of statistical machine learning, Supervised and unsupervised learning, Model evaluation and selection. Regression Analysis - Linear regression, Polynomial regression, Regularization techniques (e.g., Ridge, Lasso), Case study: Predicting housing prices

Unit 2: Classification Methods - Logistic regression, Naive Bayes classification, Decision trees and ensemble methods (e.g., Random Forest, Gradient Boosting), Case study: Spam email classification. Unsupervised Learning - Clustering algorithms (e.g., K-means, Hierarchical clustering), Dimensionality reduction techniques (e.g., PCA, t-SNE) Case study: Customer segmentation

Unit 3: Deep Learning Fundamentals - Neural networks and deep learning architectures, Convolutional Neural Networks (CNNs) for image analysis, Recurrent Neural Networks (RNNs) for sequential data, Case study: Image classification using CNNs.

Case studies

1. Sentiment Analysis of Social Media Data: Analyzing Twitter data to classify sentiment (positive, negative, neutral) using various machine learning models.
2. Image Classification with Convolutional Neural Networks: Building a model to classify images from the CIFAR-10 dataset using CNNs.
3. Movie Recommender System: Developing a recommendation engine using collaborative filtering techniques to suggest personalized movie recommendations.
4. Credit Card Fraud Detection: Building a fraud detection model using supervised learning algorithms to identify fraudulent transactions.
5. Time Series Forecasting: Predicting stock prices or weather patterns using time series forecasting techniques such as ARIMA or LSTM models.

Recommended Reference Books:

"Pattern Recognition and Machine Learning" by Christopher Bishop

"The Elements of Statistical Learning" by Trevor Hastie, Robert Tibshirani, and Jerome Friedman

"Deep Learning" by Ian Goodfellow, Yoshua Bengio, and Aaron Courville,

Course Type(**Mandatory)**			
Course Code	**Course Title**	**Credits**	**Contact Hours**
CCS/MJT/552	**Microcontroller Programming using Python**	2	**2hrs/per week**
CCS/MJP/552	Practical Based on Microcontroller Programming using Python	2	**4hrs/per week**

Course Description:

This course on Microcontroller Programming using Python is designed to provide students with a comprehensive understanding of microcontrollers and how to program them using the Python programming language. The course will cover the basics of microcontroller architecture, input/output operations, sensors and actuators interfacing, as well as advanced topics such as interrupts and communication protocols. Students will learn through a combination of theoretical knowledge, hands-on programming exercises, and case studies that highlight real-world applications of microcontroller programming.

Prerequisites:

- Basic programming knowledge (preferably Python)
- Understanding of fundamental electronics concepts (voltage, current, resistance)
- Familiarity with digital logic and binary numbering system

Course Objectives (CO):

- Introduce students to microcontroller programming using Python as a high-level language
- Provide a solid understanding of microcontroller architecture and its interaction with the external world
- Teach students how to interface sensors, actuators, and other peripheral devices with microcontrollers
- Enable students to develop and deploy real-world applications using microcontrollers and Python

Learning Outcomes (LO):

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

- Understand the fundamentals of microcontroller architecture and its programming requirements
- Write and execute Python code on microcontrollers for various applications
- Interface and communicate with different sensors, actuators, and peripherals
- Analyze and solve problems related to microcontroller programming using Python
- Design and implement microcontroller-based projects for practical applications

Course Outline:

Unit 1: Introduction to Microcontrollers - Overview of microcontrollers and their applications, Comparison of different microcontroller architectures, Introduction to Python as a programming language for microcontrollers. Microcontroller Fundamentals - Understanding microcontroller architecture, Input/output (I/O) pins and ports, Analog and digital signals, Interrupts and timers.

Unit 2: Interfacing with Microcontrollers - Serial communication (UART, SPI, I2C), Connecting sensors and actuators, Using libraries and APIs for microcontroller peripherals. Microcontroller Programming with Python - Setting up the development environment, Writing and executing Python code on microcontrollers, Working with GPIO pins and digital I/O, Reading analog signals and using analog-to-digital converters (ADCs). Implementing interrupts and timers, Connecting and reading analog and digital sensors Controlling actuators such as LEDs, motors, and displays. Data acquisition and signal processing techniques

Unit 3: Advanced Microcontroller Programming - Working with PWM (Pulse Width Modulation, Using interrupts for real-time applications, Memory management and optimization techniques, Low power modes and energy-efficient programming. Real-Time Systems and Interrupt Handling - Introduction to real-time systems and their requirements, Interrupt handling and its significance in microcontroller programming. Designing real-time systems using Python and microcontrollers. Case Studies - a) Building a temperature monitoring system using a microcontroller and a temperature sensor, b) Creating a simple home automation system with microcontrollers and actuators, c) Developing a data logging application with microcontrollers and external storage

Capstone projects

a) Designing an IoT-based Weather Monitoring System: Students will develop a weather monitoring system using a microcontroller programmed with Python. They will interface temperature and humidity sensors, collect data, and send it to a cloud platform for analysis and visualization.
b) Home Automation System: Students will create a home automation system using Python and a microcontroller. They will program the microcontroller to control various devices, such as lights and appliances, and develop a user-friendly interface for remote control.
c) Robotics Control: Students will work on a robotics project where they will use a microcontroller programmed with Python to control a robot's movement and interact with its sensors. They will develop algorithms for autonomous navigation and implement obstacle avoidance techniques.

Recommended Reference books

"Python Microcontroller Projects: Build 12 Python Microcontroller Projects with CircuitPython and Circuit Playground Express", By Donald Norris, Apress, 2021.

Semester III

Course Type(**Mandatory)**			
Course Code	**Course Title**	**Credits**	**Contact Hours**
CCS/MJT/600	**Computer Vision**	2	**2hrs/per week**
CCS/MJP/600	Practical Based on Computer Vision	2	**4hrs/per week**

Course Description:

This introductory course on practical computer vision provides students with a comprehensive foundation in the fundamental concepts, techniques, and applications of computer vision. Through a combination of theoretical lectures, hands-on coding exercises, and real-world case studies, students will gain practical knowledge and skills in various aspects of computer vision. The course will cover essential topics such as image processing, feature extraction, object detection, image segmentation, and deep learning-based approaches. Additionally, students will explore case studies that demonstrate the practical applications of computer vision in diverse fields such as healthcare, autonomous vehicles, surveillance, and augmented reality.

Prerequisites:

- Basic knowledge of programming (preferably Python)
- Familiarity with linear algebra and calculus
- Understanding of machine learning concepts

Course Objectives (CO):

- Introduce the fundamentals of computer vision and its applications
- Provide hands-on experience with popular computer vision algorithms and techniques
- Familiarize students with deep learning approaches for computer vision
- Develop practical skills in implementing computer vision systems
- Enable students to apply computer vision techniques to real-world problems

Learning Outcomes (LO):

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

- Understand the fundamental concepts and challenges of computer vision.
- Apply various pre-processing techniques to enhance images.
- Extract and describe image features using different algorithms.
- Implement object detection and recognition algorithms.
- Perform image segmentation using different methods.
- Apply deep learning techniques for computer vision tasks.
- Analyse and solve computer vision problems using appropriate algorithms.
- Demonstrate proficiency in implementing computer vision systems in Python.
- Evaluate and compare different computer vision approaches for specific tasks.
- Apply computer vision techniques to real-world case studies and projects.

Course Outline:

Unit 1: Introduction to Computer Vision - Overview of computer vision and its applications, Image formation and representation, Color models and image enhancement techniques, Image filtering and noise removal. Feature Extraction and Descriptors - Image feature types: corners, edges, blobs, Keypoint detection and feature extraction algorithms (e.g., Harris corner detection, SIFT, SURF, Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP)), Feature descriptors and matching techniques (e.g., SIFT descriptors, RANSAC), Feature tracking and optical flow

Unit 2: Object Detection, Recognition, Tracking and Motion Analysis - Introduction to object detection and localization, Traditional methods for object detection (e.g., Viola-Jones algorithm), Introduction to deep learning-based object detection (e.g., Faster R-CNN, YOLO), Tracking techniques (Kalman filter, particle filter, etc.), Multi-object tracking and data association, Optical flow estimation (Lucas-Kanade, Horn-Schunck, etc.), Motion segmentation and tracking, Event-based vision and bio-inspired motion analysis. Image Segmentation - Image segmentation techniques (e.g., thresholding, region-based segmentation, graph cuts), Introduction to semantic segmentation and instance segmentation, Deep learning-based segmentation models (e.g., U-Net, Mask R-CNN)

Unit 3: Deep Learning for Computer Vision - Introduction to convolutional neural networks (CNNs) for computer vision, Transfer learning and fine-tuning pre-trained CNN models, Training CNN models from scratch, Introduction to generative adversarial networks (GANs),

Case Studies and Applications – a) Computer vision in healthcare: Medical image analysis, disease diagnosis, b) Computer vision in autonomous vehicles: Object detection, lane detection, c) Computer vision in surveillance: Activity recognition, anomaly detection, d) Computer vision in augmented reality: Marker detection, object tracking

Recommended Reference Books:

"Computer Vision: Algorithms and Applications" by Richard Szeliski

"Deep Learning" by Ian Goodfellow, Yoshua Bengio, and Aaron Courville

"Practical Python and OpenCV" by Adrian Rosebrock

"Mastering OpenCV 4 with Python" by Alberto Fernandez Villan

Course Type(**Mandatory)**			
Course Code	**Course Title**	**Credits**	**Contact Hours**
CCS/MJT/601	**Data Mining and Warehousing with Python**	2	**2hrs/per week**
CCS/MJP/601	Practical Based on Data Mining and Warehousing with Python	2	**4hrs/per week**

Course Description:

This introductory course provides an overview of the fundamental concepts and techniques of data mining and warehousing using Python. Students will learn the basics of data mining, including data pre-processing, association rule mining, clustering, and classification. They will also explore the principles of data warehousing, including data modelling, ETL (Extract, Transform, Load) processes, and OLAP (Online Analytical Processing) techniques. The course includes hands-on exercises, case studies, and references to key books in the field.

Prerequisites:

- Basic knowledge of programming concepts
- Familiarity with Python programming language (recommended)
- Understanding of fundamental statistical concepts
- Basic knowledge of databases and SQL

Course Objectives (CO):

- Understand the concepts and principles of data mining and warehousing
- Gain proficiency in using Python for data mining and warehousing tasks
- Learn various data pre-processing techniques for preparing data for analysis

- Explore different data mining algorithms and their applications
- Understand the architecture and design principles of data warehousing
- Apply data visualization techniques to communicate insights effectively
- Analyse real-world case studies to gain practical experience in data mining and warehousing

Learning Outcomes (LO):

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

- Understand the role and significance of data mining and warehousing in decision-making processes.
- Apply Python programming language and relevant libraries for data mining and warehousing tasks.
- Pre-process and clean data effectively, including handling missing values, outliers, and data transformation.
- Apply various data mining techniques such as association rule mining, clustering, classification, regression, and text mining.
- Design and implement a data warehouse, including ETL processes and dimensional modeling.
- Utilize OLAP tools for data analysis and reporting.
- Create compelling data visualizations and interactive dashboards.
- Analyse real-world case studies to solve business problems using data mining and warehousing techniques.

Course Outline:

Unit 1: Introduction to Data Mining - Overview of data mining and its applications, Data pre-processing: cleaning, integration, transformation, and reduction, Exploratory data analysis and visualization techniques, Introduction to Python libraries for data mining (e.g., Pandas, NumPy, Matplotlib). Association Rule Mining - Apriori algorithm and association rule generation, Evaluation metrics for association rules, Practical applications and case studies of association rule mining

Unit 2: Clustering - Introduction to clustering algorithms (e.g., k-means, hierarchical clustering), Evaluation metrics for clustering, Practical applications and case studies of clustering. Classification - Introduction to classification algorithms (e.g., decision trees, Naive Bayes, logistic regression), Evaluation metrics for classification, Practical applications and case studies of classification,

Unit 3: Introduction to Data Warehousing - Overview of data warehousing and its components, Data modelling: star schema, snowflake schema, ETL processes: extraction, transformation, and loading of data, Practical applications and case studies of data warehousing. OLAP and Data Cube - Introduction to OLAP and its benefits, Data cube representation and operations, OLAP tools and techniques, Practical applications and case studies of OLAP. Case Studies – a) Retail Market Basket Analysis: Analysing customer purchase patterns to identify associations between products for targeted marketing b) Customer Segmentation: Clustering customers based on demographic and behavioural data to tailor marketing campaigns c) Loan Default Prediction: Using classification techniques to predict the likelihood of loan default based on historical data.

Reference Books:

"Data Mining: Concepts and Techniques" by Jiawei Han, Micheline Kamber, and Jian Pei.

"Python for Data Analysis" by Wes McKinney.

"Data Warehousing Fundamentals" by Paulraj Ponniah.

"The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling" by Ralph Kimball and Margy Ross.

Course Type(**Mandatory)**			
Course Code	**Course Title**	**Credits**	**Contact Hours**
CCS/MJT/602	**Internet of Things (IoT)**	2	**2hrs/per week**
CCS/MJP/602	Practical Based on Internet of Things (IoT)	2	**4hrs/per week**

Course Description:

The introductory course on Internet of Things (IoT) aims to provide students with a comprehensive understanding of IoT technologies, applications, and their impact on various industries. The course will cover the fundamental concepts, architectures, protocols, and challenges associated with IoT. It will also include case studies highlighting real-world implementations of IoT, allowing students to gain practical insights into the subject matter.

Prerequisites:

- Basic knowledge of computer networks and protocols
- Familiarity with programming concepts (preferably in a language like Python or Java)
- Understanding of data management and analytics principles

Course Objectives (CO):

- To provide an understanding of the fundamental concepts and components of the Internet of Things (IoT)
- To explore various technologies and protocols used in IoT networks and devices
- To introduce methods for acquiring, processing, and analyzing IoT data
- To examine real-world IoT applications and case studies in different domains
- To address the security and privacy challenges associated with IoT deployments
- To enable students to design and develop IoT solutions using appropriate platforms and frameworks

Learning Outcomes (LO):

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

- Define and explain the key concepts, components, and architecture of the Internet of Things
- Identify and evaluate different IoT devices, sensors, and communication technologies
- Design and implement IoT data acquisition and processing systems
- Analyze and interpret IoT data for decision-making purposes
- Develop IoT applications using relevant platforms and frameworks
- Assess security risks and implement appropriate measures for securing IoT deployments
- Analyze and discuss real-world case studies of successful IoT implementations in various domains

Course Outline:

Unit 1: Introduction to IoT - Definition, history, and evolution of IoT, Key components of an IoT system, IoT ecosystem and stakeholders. IoT Architecture and Protocols - IoT network architectures (centralized, decentralized, hybrid), Communication protocols (MQTT, CoAP, HTTP, etc.), Sensor networks and data aggregation. IoT Connectivity Technologies - Wireless communication (Wi-Fi, Bluetooth, Zigbee, LoRa, etc.), Cellular technologies (2G, 3G, 4G, 5G), Edge computing and fog computing

Unit 2: IoT Data Management and Analytics - Data collection, storage, and processing in IoT, Big Data analytics and machine learning for IoT, Data security and privacy considerations. IoT Applications in Smart Home and Cities - Smart home automation systems, Intelligent transportation systems, Energy management and environmental monitoring. IoT in Healthcare and Wearable Devices - Remote patient monitoring, Smart healthcare systems, Wearable technology and healthcare applications

Unit 3: Industrial IoT (IIoT) and Smart Manufacturing - Industrial automation and control system, Predictive maintenance and asset tracking, Supply chain management and logistics, Future Trends and Challenges in IoT - Emerging trends in IoT (AI integration, edge intelligence, etc), Ethical considerations and societal impact of IoT, Open research problems and future directions. Case Studies a) Smart agriculture and precision farming b) Smart retail and inventory management, c) Smart buildings and infrastructure d) IoT-enabled environmental monitoring

Recommended Reference Books:

"Building the Internet of Things: Implement New Business Models, Disrupt Competitors, Transform Your Industry" by Maciej Kranz

"Internet of Things: Principles and Paradigms" by Rajkumar Buyya, Amir Vahid Dastjerdi

"Designing Connected Products: UX for the Consumer Internet of Things" by Claire Rowland, Elizabeth Goodman, Martin Charlier, Ann Light

"Internet of Things for Architects: Architecting IoT solutions by implementing sensors, communication infrastructure, edge computing, analytics, and security" by Perry Lea

"The Fourth Industrial Revolution" by Klaus Schwab

Semester IV

Course Type(**Mandatory)**			
Course Code	**Course Title**	**Credits**	**Contact Hours**
CCS/MJT/650	**Artificial Intelligence**	2	**2hrs/per week**
CCS/MJP/650	Practical Based on Artificial Intelligence	2	**4hrs/per week**

Course Description:

This course is designed to provide students with a comprehensive introduction to practical applications of Artificial Intelligence (AI) in various industries. The course will cover fundamental concepts, techniques, and case studies that demonstrate how AI is being used to solve real-world problems. Students will gain hands-on experience through practical exercises and explore the ethical implications and future trends of AI.

Prerequisites:

- Basic programming knowledge (Python preferred)
- Understanding of linear algebra and calculus
- Familiarity with statistics and probability theory

Course Objectives (CO):

- Gain a practical understanding of the core concepts and techniques in Artificial Intelligence.
- Develop the skills to implement and train machine learning models for various tasks.
- Explore advanced topics in deep learning, natural language processing, computer vision, and reinforcement learning.
- Apply AI techniques to real-world problems through case studies.
- Understand the ethical considerations and challenges associated with AI implementation.

Learning Outcomes (LO):

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

- Implement and evaluate machine learning models using popular libraries and frameworks.
- Build and train neural networks for various AI tasks.
- Apply NLP techniques to analyse and process textual data.
- Utilize computer vision algorithms for image analysis and object recognition.
- Understand the principles of reinforcement learning and develop RL-based agents.
- Analyse and interpret AI case studies in different domains.
- Demonstrate an awareness of ethical considerations in AI and the impact of AI on society.

Course Outline:

Unit 1: Introduction to Artificial Intelligence - Definition and history of AI, Types of AI systems, AI applications and impact on society. Revisiting Machine Learning Fundamentals - Supervised, unsupervised, and reinforcement learning, Feature engineering and data pre-processing, Evaluation metrics

Unit 2: Neural Networks and Deep Learning - Introduction to neural networks, Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Deep learning frameworks (e.g., TensorFlow, PyTorch), Reinforcement Learning - Markov Decision Processes (MDPs), Q-learning and policy gradients, Deep reinforcement learning.

Unit 3: Future Trends and Career Opportunities in AI - Emerging AI technologies (e.g., Generative AI, Quantum AI) , AI Case studies – a) Natural Language Processing (NLP) - Sentiment analysis and language generation, b) Computer Vision and Image Processing - Autonomous vehicles, c) Recommender Systems - Personalized recommendation systems, d) AI in Healthcare - AI-assisted diagnosis, e) AI in Manufacturing and Logistics - AI in manufacturing processes, f) AI in Education - Personalized education with AI, f) AI in Governance

Recommended Reference Books:

"Artificial Intelligence: A Modern Approach" by Stuart Russell and Peter Norvig

"Deep Learning" by Ian Goodfellow, Yoshua Bengio, and Aaron Courville

"Natural Language Processing with Python" by Steven Bird, Ewan Klein, and Edward Loper

"Computer Vision: Algorithms and Applications" by Richard Szeliski

"Reinforcement Learning: An Introduction" by Richard S. Sutton and Andrew G. Barto

"Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy" by Cathy O'Neil

"Superintelligence: Paths, Dangers, Strategies" by Nick Bostrom

"The Hundred-Page Machine Learning Book" by Andriy Burkov

"Prediction Machines: The Simple Economics of Artificial Intelligence" by Ajay Agrawal, Joshua Gans, and Avi Goldfarb

"Machine Learning Yearning" by Andrew Ng (available online for free)

Course Type(**Mandatory)**			
Course Code	**Course Title**	**Credits**	**Contact Hours**
CCS/MJT/651	**Data Analytics**	2	**2hrs/per week**
CCS/MJP/651	Practical Based on Data Analytics	2	**4hrs/per week**

Course Description:

This course is designed to provide participants with practical knowledge and skills in data analytics, enabling them to effectively analyse and derive insights from real-world data. The course combines theoretical concepts, hands-on exercises, case studies, and reference books to create a comprehensive learning experience. Participants will gain a solid foundation in data analytics techniques, learn to work with various data analysis tools, and apply their knowledge through practical case studies.

Prerequisites:

- Basic knowledge of statistics and probability
- Familiarity with programming concepts (preferably Python or R)
- Understanding of database concepts and SQL
- Some exposure to data analysis or data-driven decision-making

Course Objectives (CO):

- To provide students with a comprehensive understanding of data analytics concepts, techniques, and tools.
- To develop practical skills for data collection, pre-processing, and exploratory analysis.

- To introduce students to various data mining techniques and their applications in solving real-world problems.
- To equip students with the knowledge and skills required for predictive modelling and machine learning in data analytics.
- To explore big data analytics concepts and scalable data processing frameworks.
- To enhance students' ability to apply data analytics techniques to real-world case studies and projects.
- To promote ethical considerations and responsible use of data in the context of data analytics.

Learning Outcomes (LO):

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

- Understand the fundamental concepts and principles of data analytics.
- Collect, preprocess, and analyze data using various techniques and tools.
- Apply exploratory data analysis techniques to gain insights and visualize data effectively.
- Utilize different data mining techniques for pattern discovery, clustering, and classification.
- Build predictive models using appropriate algorithms and evaluate their performance.
- Apply machine learning algorithms to solve data analytics problems.
- Understand the challenges and opportunities associated with big data analytics.
- Develop the ability to critically analyze and solve real-world data analytics problems.
- Demonstrate ethical considerations and responsible use of data in the context of data analytics.

Course Outline

Unit 1: Introduction to Data Analytics - Understanding the importance of data analytics, Overview of data analytics lifecycle, Introduction to data analysis tools and technologies. Data Preparation and Cleaning - Data collection and pre-processing techniques, Handling missing data and outliers, Data transformation and feature engineering. Exploratory Data Analysis (EDA) - Descriptive statistics and data visualization, Univariate and multivariate analysis techniques, Exploring relationships and patterns in data.

Unit 2: Statistical Analysis - Hypothesis testing and confidence intervals, Parametric and non-parametric tests, Regression analysis and correlation. Predictive Analytics - Introduction to predictive modelling, Regression and classification algorithms, Model evaluation and selection. Time Series Analysis - Understanding time series data, Trend analysis and seasonality, Forecasting techniques. Clustering and Dimensionality Reduction - Unsupervised learning algorithms, K-means clustering and hierarchical clustering, Principal Component Analysis (PCA)

Unit 3: Case Studies a) Text Mining and Sentiment Analysis - Basics of text mining, Text pre-processing and feature extraction, Sentiment analysis techniques, b) Network Analysis - Introduction to network theory, Social network analysis, Network visualization and metrics, c) Analysing real-world datasets and scenarios (data can be downloaded from https://www.kaggle.com/datasets), d) Applying data analytics techniques to solve problems on the Kaggle Data set (https://www.kaggle.com/datasets), e) Discussion and interpretation of case study results

Reference Book

"Data Analytics for Beginners" by John Smith

"Data Science for Business" by Foster Provost and Tom Fawcett

"Python for Data Analysis" by Wes McKinney

"Statistics for Data Science" by James Miller

"Predictive Analytics: The Power to Predict Who Will Click, Buy, Lie, or Die" by Eric Siegel

"Forecasting: Principles and Practice" by Rob J. Hyndman and George Athanasopoulos

"Pattern Recognition and Machine Learning" by Christopher M. Bishop

"Text Mining: Applications and Theory" by Michael W. Berry and Jacob Kogan

"Network Science" by Albert-László Barabási

Course Type(**Mandatory)**			
Course Code	**Course Title**	**Credits**	**Contact Hours**
CCS/MJT/652	**IoT and Cloud : Concepts, Case Studies and Applications**	2	**2hrs/per week**
CCS/MJP/652	Practical Based on IoT and Cloud : Concepts, Case Studies and Applications	2	**4hrs/per week**

Course Overview:

The IoT and Cloud course provides a comprehensive understanding of the Internet of Things (IoT) and its integration with cloud computing. Students will learn the fundamental concepts, architectures, protocols, and technologies behind IoT and explore how cloud computing enables scalable, secure, and efficient IoT deployments. The course will also include real-world case studies and reference books to provide practical insights and hands-on experience.

Prerequisites:

- Basic knowledge of computer networks and protocols
- Understanding of programming concepts and languages
- Familiarity with basic cloud computing principles
- Some background in data management and analytics

Course Objectives (CO):

- Introduce students to the fundamental concepts and principles of IoT and Cloud Computing.
- Provide an understanding of the technologies, protocols, and infrastructure used in IoT and Cloud environments.
- Explore the integration of IoT devices with cloud platforms and analyze real-world case studies.
- Address the security challenges and best practices for IoT and Cloud deployments.
- Foster critical thinking and problem-solving skills through hands-on exercises and case study analysis.
- Highlight emerging trends and future directions in IoT and Cloud Computing.

Learning Outcomes (LO):

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

- Explain the key concepts, principles, and architectures of IoT and Cloud Computing.
- Identify and evaluate various IoT technologies, protocols, and cloud service models.
- Design and implement IoT and Cloud integration solutions for specific applications.
- Analyze case studies of real-world IoT and Cloud deployments and draw insights from them.
- Apply security measures and best practices to mitigate risks in IoT and Cloud environments.
- Demonstrate proficiency in using cloud-based platforms and tools for IoT applications.
- Recognize emerging trends and future directions in the field of IoT and Cloud Computing.

Course Outline:

Unit 1: Introduction to IoT and Cloud Computing - Definition and significance of IoT and cloud computing, Evolution and growth of IoT and cloud technologies, IoT and cloud convergence. IoT Architecture and Technologies - IoT ecosystem and components, Sensor networks and connectivity protocols, Edge computing in IoT, Data acquisition and processing. Cloud Computing Fundamentals - Cloud service models (IaaS, PaaS, SaaS), Cloud deployment models (public, private, hybrid), Virtualization and containerization, Cloud security and privacy considerations.

Unit 2: IoT Data Management and Analytics - Data storage and retrieval in the cloud, Real-time data processing and analytics, Machine learning and AI for IoT, Data visualization and dashboards. Cloud-based IoT Platforms and Services - IoT platform architectures and features, Device management and provisioning, Cloud-based IoT application development, Integration with cloud APIs and services

Unit 3: Security and Privacy in IoT and Cloud - Threats and vulnerabilities in IoT systems, Authentication and access control, Data encryption and secure communication, Privacy concerns and regulatory frameworks. Emerging Trends and Future Directions - Edge AI and Fog Computing, 5G and IoT, Introduction to Blockchain in IoT, Edge-to-cloud integration.

Case Studies and Industry Applications – a) Smart homes and buildings, b) Industrial IoT and Industry 4.0, c) Healthcare and wearable devices d) Smart cities and infrastructure.

Recommended Reference Books:

"Building the Internet of Things: Implement New Business Models, Disrupt Competitors, Transform Your Industry" by Maciej Kranz

"IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things" by David Hanes and Gonzalo Salgueiro

"Cloud Computing: Concepts, Technology & Architecture" by Thomas Erl, Ricardo Puttini, and Zaigham Mahmood

"Internet of Things for Architects: Architecting IoT Solutions by Implementing Sensors, Communication Infrastructure, Edge Computing, Analytics, and Security" by Perry Lea

"Internet of Things (A Hands-on-Approach)" by Arshdeep Bahga and Vijay Madisetti

Course Contents for Skill / Advance Course (Programming Elective Group Basket)

Programming Group	**Programming - 1** CCS/MJP/503-507	**Programming-2** CCS/MJP/553-557	**Programming -3** CCS/MJP/603-607
Java Group	Core Java	Advance Java	Android
Open Group	Python	Advanced Python	Open Web Programming (PHP)
Web Scripting Group	VB & JavaScript	NodeJS	React

Note: *,# Student is advised to select the any one course from the pool of courses, however horizontal selection of courses to be followed at the time of selection of the course.

Programming Group: Java Group

Course Type(**Mandatory)**			
Course Code	**Course Title**	**Credits**	**Contact Hours**
CCS/MJP/503	**Core Java**	2	**4hrs/per week**

Course Description:

This course is designed to provide a comprehensive understanding of Core Java programming concepts and best practices. It covers essential topics from basic syntax and control flow to object-oriented programming, exception handling, and file I/O. The course incorporates case studies to reinforce learning and includes a list of recommended reference books for further exploration.

Prerequisite:

No prior programming experience is required for this course. However, a basic understanding of computer systems and general familiarity with using computers is recommended.

Course Objectives (CO):

- Understand the fundamental concepts and principles of the Java programming language.
- Gain proficiency in writing, compiling, and executing Java programs.
- Develop object-oriented programming skills and design reusable code.
- Familiarize yourself with the Java Collections Framework and its various data structures.
- Acquire knowledge of GUI programming using Swing.
- Explore multithreading and concurrent programming concepts.

Learning Outcome (LO):

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

- Write Java programs to solve simple to complex programming problems.
- Understand and apply object-oriented programming concepts effectively.
- Utilize the Java Collections Framework to manage and manipulate data efficiently.
- Develop interactive graphical user interfaces using Swing.
- Implement multithreading and concurrent programming techniques.
- Analyze and solve programming challenges using Java.

Course Outline:

Unit 1: Introduction to Java - History and features of Java, Java development environment setup, Basic syntax, data types, and variables, Operators and expressions, Control flow statements (if-else, loops), Object-Oriented Programming (OOP) Basics - Classes and objects, Encapsulation, Inheritance, and Polymorphism, Method overloading and overriding, Constructors and static members, Access modifiers

Unit 2: Advanced OOP Concepts - Abstract classes and interfaces, Packages and access control, Exception handling, Enumerations, Inner classes, Java Collections Framework - Introduction to collections, ArrayList, LinkedList, and Vector, HashSet, LinkedHashSet, and TreeSet, HashMap, LinkedHashMap, and TreeMap, Iterators and foreach loop, File Handling and I/O - File class and file operations, Character streams and byte streams, Reading and writing files, Serialization and deserialization, Working with directories, Multithreading - Understanding threads, Creating and running threads, Synchronization and thread safety, Thread communication and coordination, Thread pools and executors

Unit 3: GUI Programming with Applet and Swing – Life Cycle of Applets & Swings components, Creating UI components – Applet & Swings (buttons, labels, text fields, etc.), Event-driven programming, Layout managers, Creating Dialogs and message boxes, Swing Layouts. Revisiting to Databases and SQL - Understanding databases and database management systems, Introduction to SQL (Structured Query Language), Performing basic database operations using SQL. Database Integration with Java (JDBC) - Overview of JDBC (Java Database Connectivity), Connecting to databases using JDBC, Executing SQL queries and retrieving results in Java, Advanced JDBC Concepts -Prepared statements and parameterized queries, Batch processing and transaction management, Handling result sets and metadata

Case Studies and Project Work - Case studies covering real-world scenarios, implementing case study solutions using Core Java. Following Case studies are required to be addressed by the students during their laboratory work.

> Case Study a) Building a Contact Management System - Analyzing requirements, Designing the database schema, Implementing the system using Core Java, databases, and Swing, b) Creating an Inventory Management Application - Understanding the business case, Designing the database structure Developing the application with Core Java, databases, and Swing
>
> Capstone Project: Final Capstone project work incorporating multiple concepts learned by the student during the course work with Code optimization and best practices.

Recommended Reference Books:

"Head First Java" by Kathy Sierra and Bert Bates

"Core Java, Volume I -- Fundamentals" by Cay S. Horstmann

"Effective Java" by Joshua Bloch

"Java: The Complete Reference" by Herbert Schildt

"Thinking in Java" by Bruce Eckel

Course Type(**Mandatory)**

Course Code	Course Title	Credits	Contact Hours
CCS/MJP/553	**Advance Java**	2	**4hrs/per week**

Course Description:

This course provides practical and advanced training in Java programming with a focus on building web applications using Servlets, Struts, and Hibernate frameworks. Students will gain hands-on experience in developing robust and scalable Java-based web applications by utilizing these powerful frameworks. The course emphasizes best practices, design patterns, and industry-standard techniques to create efficient and maintainable web applications.

Prerequisite:

- Solid understanding of core Java programming concepts
- Familiarity with web development basics (HTML, CSS, and JavaScript)
- Knowledge of relational databases and SQL

Course Objectives (CO):

- Develop a deep understanding of advanced Java web development concepts
- Gain proficiency in using Servlets, Struts, and Hibernate frameworks
- Learn to design and build scalable and maintainable web applications
- Acquire hands-on experience in implementing MVC architecture
- Develop skills in database integration and ORM techniques

Learning Outcomes (LO):

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

- Design and develop advanced Java-based web applications
- Effectively use Servlets, Struts, and Hibernate frameworks in web development projects
- Apply industry-standard design patterns and best practices
- Implement secure authentication and session management
- Create efficient database-driven web applications
- Demonstrate proficiency in integrating and utilizing multiple frameworks

Course Outline:

Unit 1: Introduction to Advanced Java Web Development - Overview of Java Servlets, Struts, and Hibernate, Comparison of different web frameworks, Understanding the MVC (Model-View-Controller) architecture. Servlets: Building Dynamic Web Applications - Servlet life cycle and request handling, Handling form data and request parameters, Session management and authentication, Servlet filters and listeners, Error handling and exception management

Unit 2: Struts Framework: Structured Web Application Development - Introduction to Struts framework and its components, Configuring Struts and defining action mappings, working with forms and validation, Managing database operations using Struts, Implementing security and authentication in Struts. Hibernate: Object-Relational Mapping (ORM) - Introduction to Hibernate and ORM concepts, Configuring Hibernate with different database systems, Mapping Java objects to database tables, Performing CRUD operations using Hibernate, Querying data using Hibernate Query Language (HQL)

Unit 3: Integration of Servlets, Struts, and Hibernate - Leveraging the power of Servlets, Struts, and Hibernate together, building a complete end-to-end web application, Implementing layered architecture and separation of concerns, Handling transactions and database operations. Case Studies a) Building an e-commerce platform using Servlets, Struts, and Hibernate, b) Developing a social networking application with advanced Java web technologies, c) Creating a banking system with secure authentication and transaction handling, d) Implementing a content management system (CMS) using Java web frameworks

Recommended Reference Books:

"Head First Servlets and JSP" by Bryan Basham, Kathy Sierra, and Bert Bates

"Struts 2 in Action" by Don Brown, Chad Davis, Scott Stanlick, and Ted Husted

"Hibernate in Action" by Christian Bauer and Gavin King

"Pro Spring MVC: With Web Flow" by Marten Deinum, Koen Serneels, and Colin Yates

"Java Persistence with Hibernate" by Christian Bauer and Gavin King

Course Type(**Mandatory)**			
Course Code	**Course Title**	**Credits**	**Contact Hours**
CCS/MJP/603	**Android Programming**	2	**4hrs/per week**

Course Description:

This course is designed to provide a comprehensive understanding of Android programming, starting from the basics and progressing towards more advanced concepts. Students will learn the fundamentals of Android app development, including user interfaces, data storage, networking, and multimedia integration. The course will also incorporate case studies and practical examples to demonstrate real-world applications of Android programming. Students will gain hands-on experience by developing their own Android apps and exploring reference books to deepen their knowledge.

Prerequisite:

- Basic knowledge of Java programming language
- Familiarity with object-oriented programming concepts

Course Objectives (CO):

- Understand the fundamental concepts and architecture of the Android platform.
- Gain proficiency in developing Android applications using Java.
- Acquire knowledge of various Android components and their functionalities.
- Learn to design user-friendly and visually appealing Android interfaces.
- Develop skills in handling data storage, network communication, and multimedia integration in Android applications.
- Apply best practices and coding standards for developing high-quality Android apps.
- Gain hands-on experience by working on real-world case studies.

Learning Outcome (LO):

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

- Design and develop functional Android applications from scratch.
- Implement key Android features such as activities, services, and content providers.
- Create intuitive user interfaces using XML layouts and UI components.
- Perform network communication and data persistence in Android apps.
- Integrate multimedia elements and utilize device sensors.
- Employ advanced Android techniques such as notifications, maps, and external API integration.
- Apply the knowledge gained from case studies to build complex Android applications.

Course Outline:

Unit 1: Introduction to Android Development - Overview of Android ecosystem, setting up development environment (Android Studio, SDK, emulators), Understanding the Android project structure, Introduction to Android components (activities, fragments, services). User Interface Development - Layouts and views, User input and event handling, working with menus and dialogues, Creating responsive and adaptive interfaces, Material Design principles and guidelines

Unit 2: Data Storage and Persistence - Using SQLite database, Content Providers and data sharing SharedPreferences for app preferences, Working with files and external storage, Introduction to cloud storage and synchronization (Firebase). Networking and Web Services - Making HTTP requests (HTTP libraries, RESTful APIs), Parsing JSON and XML data, Working with web sockets, Authentication and authorization mechanisms, Caching and offline capabilities

Unit 3: Multimedia Integration - Working with images, audio, and video, integrating camera and gallery functionalities, playing media files and streaming, Implementing notifications and push notifications, Location-based services and maps integration, Testing and debugging Android applications. Case Studies and Project Development - Exploring case studies of popular Android applications (Data Collected), Analysing and dissecting their architecture and key features (Social Media Applications), Implementing key components of case study apps, Applying best practices and optimization techniques,

Recommended Reference Books:

"Android Programming: The Big Nerd Ranch Guide" by Bill Phillips, Chris Stewart, and Kristin Marsicano

"Head First Android Development" by Dawn Griffiths and David Griffiths

"Android Studio 4.0 Development Essentials - Kotlin Edition" by Neil Smyth

"Android App Development for Dummies" by Michael Burton

"Professional Android 4 Application Development" by Reto Meier

Programming Group: Open Group

Course Type(**Mandatory)**			
Course Code	**Course Title**	**Credits**	**Contact Hours**
CCS/MJP/505	**Python Programming**	2	**4hrs/per week**

Course Description:

This course is designed to provide students with a comprehensive understanding of Python programming. It covers the fundamentals of Python, including syntax, data types, control structures, functions, and object-oriented programming concepts. Through a combination of lectures, hands-on exercises, and case studies, students will gain practical experience in solving real-world problems using Python.

Prerequisites:

- There are no specific prerequisites for this course. However, a basic understanding of programming concepts and familiarity with any programming language would be beneficial.

Course Objectives (CO):

- To introduce students to the fundamentals of Python programming language.
- To develop students' problem-solving skills using Python.
- To enable students to apply Python programming techniques to solve real-world problems.
- To provide hands-on experience in Python programming through practical exercises and case studies.
- To prepare students for further studies or careers in software development, data analysis, or scientific computing.

Learning Outcomes (LO):

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

- Understand and apply the core concepts of Python programming.

- Write Python programs to solve a variety of computational problems.
- Design and implement object-oriented programs in Python.
- Utilize Python libraries and modules for specific tasks.
- Analyse and debug Python code for errors and exceptions.
- Develop a basic understanding of GUI programming using Tkinter.

Course Outline:

Unit 1: Introduction to Python - History and features of Python, Installing Python and setting up the development environment, Basic Python syntax and data types, Variables, operators, and expressions. Control Structures, Conditional statements (if, else, elif), Looping statements (for, while), Control flow and branching

Unit 2: Functions and Modules - Defining and calling functions, Function parameters and return values, Recursion, Introduction to modules and libraries, Data Structures - Lists, tuples, and dictionaries, String manipulation, Sets and frozen sets, Working with files and directories

Unit 3: Regular expressions, File handling and input/output operations, Debugging and error handling Introduction to GUI programming with Tkinter

Case Studies: Throughout the course, students will work on case studies that demonstrate the practical application of Python programming in various domains, such as: a) Analyse and visualizing data using Python libraries like NumPy, Pandas, and Matplotlib, b) Building web applications with frameworks like Django or Flask., c) Implementing algorithms for machine learning and data mining, d) Automating tasks and working with APIs., e) Creating graphical user interfaces (GUI) for desktop applications.

Recommended Reference Book:

"Python Programming: An Introduction to Computer Science", John Zelle, Franklin, Beedle & Associates Inc, 2016

"Fluent Python" by Luciano Ramalho

"Python Cookbook" by David Beazley and Brian K. Jones

"Python Crash Course" by Eric Matthes

"Python for Data Analysis" by Wes

Course Type(**Mandatory)**			
Course Code	**Course Title**	**Credits**	**Contact Hours**
CCS/MJP/555	**Advance Python Programming**	2	**4hrs/per week**

Course Description:

The Advanced Python Programming course is designed for individuals who have a solid foundation in Python and want to enhance their skills to an advanced level. This course focuses on advanced concepts, techniques, and best practices that will enable students to develop robust and efficient Python applications. Through a combination of lectures, hands-on exercises, and case studies, participants will deepen their understanding of Python and learn how to leverage its full potential.

Prerequisites:

- Proficiency in Python programming language, including knowledge of basic syntax, data types, control structures, and functions.
- Familiarity with object-oriented programming concepts.

Course Objectives (CO):

- Develop a deep understanding of advanced Python concepts and techniques.

- Master the design and implementation of complex data structures and algorithms in Python.
- Gain proficiency in object-oriented programming and apply design patterns effectively.
- Acquire knowledge of functional programming paradigms and utilize them in Python.
- Learn how to write Pythonic code and follow best practices for code organization and style.
- Understand concurrency and parallelism in Python and apply them to optimize performance.
- Enhance debugging and testing skills to ensure the quality of Python applications.

Learning Outcomes (LO):

By the end of this course, participants will be able to:

- Design and implement advanced data structures and algorithms in Python.
- Utilize object-oriented programming principles to develop modular and reusable code.
- Apply functional programming concepts to write elegant and concise Python code.
- Implement concurrency and parallelism in Python to improve performance.
- Write clean, maintainable, and Pythonic code following best practices.
- Effectively debug and test Python applications for identifying and fixing issues.
- Apply the learned concepts to real-world case studies and solve complex problems.

Course Outline:

Unit 1: Advanced Data Structures and Algorithms in Python - Advanced data structures: sets, dictionaries, heaps, and graphs, Algorithm design and analysis: recursion, sorting, searching, and dynamic programming, Time and space complexity analysis

Unit 2: Object-Oriented Programming (OOP) in Python - Inheritance, polymorphism, and encapsulation, Design patterns and their implementation in Python, Advanced OOP concepts: abstract classes, interfaces, and multiple inheritance, Functional Programming in Python - Higher-order functions, lambda expressions, and closures, Immutable data structures and pure functions, Functional programming techniques: map, filter, reduce, and recursion

Unit 3: Concurrency and Parallelism in Python - Multithreading and multiprocessing, Synchronization and thread safety, Asynchronous programming with async/await. Python Database Libraries - Overview of popular Python database libraries (e.g., SQLAlchemy, psycopg2), Installation and setup of database libraries, connecting to databases using Python. Database Querying with Python - Executing SQL queries using Python, Fetching and manipulating query results, Parameterized queries and prepared statements, Object-Relational Mapping (ORM) - Introduction to ORM frameworks (e.g., SQLAlchemy), Mapping database tables to Python classes, Performing CRUD operations using ORM

Case Studies: Throughout the course, students will analyse and work on various case studies to apply the advanced Python concepts they have learned. The case studies will cover domains such as data analysis, web development, scientific computing, and machine learning. These real-world scenarios will provide practical experience and enable participants to tackle complex problems using advanced Python programming techniques.

Recommended Reference Book:

"Python Cookbook", David Beazley and Brian K. Jones, O'Reilly Media, 2018

"Fluent Python" by Luciano Ramalho

"Python Cookbook" by David Beazley and Brian K. Jones

"Python Crash Course" by Eric Matthes

"Python for Data Analysis" by Wes

Course Type(**Mandatory**)			
Course Code	**Course Title**	**Credits**	**Contact Hours**
CCS/MJP/605	**Open Web Programming (PHP)**	2	**4hrs/per week**

Course Description:

This course is designed to provide students with a comprehensive understanding of PHP programming language, along with practical case studies. Students will learn the fundamentals of PHP, including its syntax, data types, control structures, functions, and object-oriented programming concepts. The course will also cover advanced topics such as database integration, web application development, security considerations, and performance optimization techniques.

Prerequisite:

- Basic understanding of programming concepts
- Familiarity with HTML, CSS, and JavaScript is preferred but not mandatory

Course Objectives (CO):

- To introduce students to the fundamentals of PHP programming language
- To enable students to develop dynamic web applications using PHP
- To provide hands-on experience with database integration and web security in PHP
- To enhance students' problem-solving and critical thinking skills in the context of PHP development

Learning Outcome (LO):

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

- Write PHP code to solve programming problems and develop web applications
- Integrate PHP with databases for efficient data management
- Implement security measures to protect web applications from common vulnerabilities
- Optimize PHP code and database queries for improved performance
- Analyze and troubleshoot PHP applications using debugging tools and techniques

Course Outline:

Unit 1: Introduction to PHP - Basics of PHP syntax, Variables and data types, Control structures (if-else, loops), Functions and array, PHP Programming Fundamentals - File handling and input/output operations, String manipulation, Regular expressions, Error handling and debugging

Unit 2: Object-Oriented Programming in PHP - Classes and objects, Inheritance and polymorphism, Encapsulation and data abstraction, Exception handling. Database Integration - Introduction to databases (MySQL, SQLite, etc.), SQL queries and database operations, Connecting PHP with databases, CRUD operations (Create, Read, Update, Delete).

Unit 3. Web Application Development - Basics of web development (HTML, CSS, JavaScript), Server-side scripting and client-server communication, Handling form (GET, POST) submissions, Session management and cookies. Security Considerations - Common web vulnerabilities (cross-site scripting, SQL injection), Input validation and data sanitization, Password hashing and encryption, User authentication and authorization. Performance Optimization - Caching techniques, Code profiling and optimization, Database query optimization, Load balancing and scalability considerations

Recommended Reference Book:

"PHP and MySQL Web Development" by Luke Welling and Laura Thomson.

Programming Group: Web Scripting Group

Course Type(**Mandatory)**			
Course Code	**Course Title**	**Credits**	**Contact Hours**
CCS/MJP/507	**VB & Java Script**	2	**4hrs/per week**

Course Description:

This three-credit course provides an in-depth exploration of Visual Basic (VB) and JavaScript programming languages. Students will learn the fundamental concepts, syntax, and best practices of both languages, enabling them to develop interactive and dynamic web and desktop applications. The course includes hands-on exercises, case studies, and reference books to enhance the learning experience and promote practical application of the concepts learned.

Prerequisite:

No prior programming experience is required for this course. However, a basic understanding of computer concepts and familiarity with using computers and the internet is recommended.

Course Objectives (CO):

- To introduce students to the fundamentals of VB and JavaScript programming languages.
- To enable students to create interactive web pages and develop desktop applications using VB.
- To familiarize students with the integration of VB and JavaScript for enhanced functionality.
- To develop problem-solving skills through hands-on case studies and practical assignments.
- To prepare students for further studies or careers in web development and software engineering.

Learning Outcomes (LO):

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

- Understand and apply the syntax, concepts, and principles of VB and JavaScript programming.
- Develop web pages with interactive elements and dynamic content using JavaScript.
- Design and implement desktop applications using VB, incorporating JavaScript functionality.
- Solve programming problems using logical thinking and debugging techniques.
- Apply VB and JavaScript programming knowledge to real-world scenarios through case studies.

Course Outline:

Unit 1: Introduction to Visual Basic Programming - Overview of Visual Basic programming language, Introduction to the Integrated Development Environment (IDE), Basic syntax and data types in Visual Basic, Variables, operators, and expressions, Control structures: decision-making and looping, Arrays and collections. Advanced Visual Basic Programming - Object-oriented programming concepts in Visual Basic, Classes, objects, and inheritance, Exception handling and error trapping, File handling and data input/output, User interface design with forms and controls, Event-driven programming

Unit 2: Introduction to JavaScript Programming - Introduction to JavaScript and its role in web development, JavaScript syntax, variables, and data types, Control flow and conditional statements, Functions and scope, Working with arrays and objects, DOM manipulation and event handling. Advanced JavaScript Programming- Advanced JavaScript concepts: closures, prototypes, and modules, Asynchronous programming with JavaScript, Error handling and debugging techniques, working with JSON and AJAX, Introduction to modern JavaScript frameworks (e.g., React, Angular)

Unit 3: Case Studies: a) Building a Desktop Application with Visual Basic: Students will develop a desktop application using Visual Basic, incorporating various concepts covered in the course. The case study will focus on user interface design, data management, and implementing business logic b) Creating Dynamic Web Pages with JavaScript: Students will create interactive web pages using JavaScript. They will learn to manipulate the

Document Object Model (DOM), handle events, and retrieve data from external sources, showcasing the power of JavaScript in web development.

Recommended Reference Books:

"Visual Basic 2019 in 24 Hours, Sams Teach Yourself" by James Foxall

"Murach's JavaScript and jQuery" by Zak Ruvalcaba and Mary Delamater

"Eloquent JavaScript: A Modern Introduction to Programming" by Marijn Haverbeke

"JavaScript: The Good Parts" by Douglas Crockford

"Professional Visual Basic 2019 and .NET Core 3.0" by Bill Sheldon, Billy Hollis, and Rob Windsor

Course Type(**Mandatory)**			
Course Code	**Course Title**	**Credits**	**Contact Hours**
CCS/MJP/557	**Node.js Programming**	2	**4hrs/per week**

Course Description:

The Node.js Programming Credit Foundation Course is designed to provide students with a comprehensive understanding of Node.js, a popular server-side JavaScript runtime environment. This course will cover the fundamental concepts, principles, and best practices of Node.js programming, enabling students to build efficient and scalable web applications. Through a combination of theoretical knowledge, practical exercises, case studies, and recommended reference books, students will gain the skills necessary to become proficient in Node.js development.

Prerequisite:

- Basic knowledge of JavaScript programming language
- Familiarity with web development concepts (HTML, CSS, and client-side JavaScript)
- Understanding of fundamental concepts of server-client architecture

Course Objectives (CO):

- Gain a thorough understanding of Node.js and its key features
- Develop skills to build server-side applications using Node.js
- Learn best practices for asynchronous programming and handling I/O operations
- Acquire knowledge of web development with Node.js, including routing, middleware, and database integration
- Explore advanced topics such as real-time communication, microservices, and performance optimization
- Analyze and comprehend real-world case studies to apply Node.js effectively in various scenarios

Learning Outcomes (LO):

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

- Build robust web applications using Node.js and related frameworks
- Implement asynchronous programming techniques to handle concurrent operations efficiently
- Integrate databases and implement user authentication in Node.js applications
- Apply best practices for testing, debugging, and deployment of Node.js applications
- Analyse and understand the architecture and implementation of real-world Node.js projects
- Solve complex programming challenges using the knowledge gained during the course

Course Outline:

Unit 1: Introduction to Node.js - Overview of Node.js and its features, Understanding the event-driven, non-blocking I/O model, Installing Node.js and setting up the development environment, Introduction to npm (Node Package Manager). JavaScript Fundamentals - Revision of JavaScript essentials, Asynchronous programming concepts and callbacks, Promises and async/await for handling asynchronous operations

Unit 2: Building Web Servers with Node.js - Creating a basic HTTP server using the built-in HTTP module, Routing and handling different types of requests, Working with Express.js, a popular Node.js web application framework. Working with Databases - Overview of database systems and their role in web applications, Introduction to MongoDB and NoSQL databases, Using MongoDB with Node.js and Mongoose ODM (Object-Data Mapping)

Unit 3: Asynchronous Programming in Node.js - Understanding the Event Loop and non-blocking I/O, Using callbacks, Promises, and async/await for asynchronous programming, Error handling and best practices, Middleware and Authentication - Implementing middleware for request processing, User authentication and authorization techniques, Working with JSON Web Tokens (JWT) for secure authentication. Real-time Applications with Socket.io, Introduction to real-time communication, Building real-time web applications with Socket.io, Broadcasting and handling events in real-time Case Studies a) Analyzing and implementing real-world Node.js applications, b) Examining scalability, performance optimization, and best practices, Case studies may include chat applications, API servers, and more

Recommended Reference Books:

"Node.js Web Development" by David Herron

"Learning Node.js: A Hands-On Guide to Building Web Applications in JavaScript" by Marc Wandschneider

"Node.js Design Patterns" by Mario Casciaro and Luciano Mammino

"Mastering Node.js" by Sandro Pasquali

"Node.js in Action" by Mike Cantelon, Marc Harter, T.J. Holowaychuk, and Nathan Rajlich

Course Type(**Mandatory)**			
Course Code	**Course Title**	**Credits**	**Contact Hours**
CCS/MJP/607	**React Programming**	2	**4hrs/per week**

Course Description:

The React Programming Course is designed to provide students with a comprehensive understanding of React, one of the most popular JavaScript libraries for building user interfaces. This course aims to equip students with the fundamental knowledge and skills necessary to develop web applications using React. Through a combination of theoretical concepts, practical exercises, case studies, and reference books, students will gain hands-on experience in creating interactive and responsive UI components.

Prerequisites:

- Basic knowledge of HTML, CSS, and JavaScript
- Familiarity with web development concepts and principles

Course Objectives (CO):

- Gain a thorough understanding of React and its core concepts
- Learn how to build interactive and dynamic user interfaces using React components
- Develop proficiency in managing state and handling events in React
- Understand the fundamentals of React Router for building single-page applications
- Acquire knowledge of popular styling approaches and libraries for React
- Learn form handling and validation techniques in React
- Explore advanced concepts such as React hooks and performance optimization

Learning Outcomes (LO):

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

- Build responsive and interactive web applications using React
- Develop reusable and modular React components
- Implement efficient state management and event handling in React applications
- Create single-page applications with React Router
- Apply styling techniques to enhance the visual appeal of React applications
- Implement form handling and validation in React
- Understand and apply advanced React concepts for optimal performance and error handling

Course Outline:

Unit 1: Introduction to React - Overview of React and its key features, Understanding the React component model Setting up a development environment for React, Introduction to JSX (JavaScript XML). React Components and State Management - Creating functional and class components, understanding component lifecycle methods, managing component state with hooks and context, Handling events and data binding in React

Unit 2: React Routing and Navigation - Introduction to React Router for handling client-side routing, Implementing navigation and nested routes in React, Managing route parameters and query strings, Implementing dynamic routing and code splitting, Styling in React - Styling options in React: CSS, inline styles, CSS modules, Using popular CSS-in-JS libraries (e.g., styled-components), Best practices for organizing and managing styles in React

Unit 3: Advanced React Concepts - State management with Redux or MobX, integrating third-party libraries and APIs with React, Server-side rendering with React (Next.js), Testing and debugging React applications. React Forms and Validation: Controlled and uncontrolled components, Form handling and validation techniques, Form libraries and formik integration, Advanced React Concepts: React hooks for custom logic (useReducer, useContext, etc.), Performance optimization techniques, Error handling and debugging in React

Recommended Reference Books:

"Learning React: Modern Patterns for Developing React Apps" by Alex Banks and Eve Porcello

"React Up and Running: Building Web Applications" by Stoyan Stefanov

"Pro React 16" by Adam Freeman

"Fullstack React: The Complete Guide to ReactJS and Friends" by Anthony Accomazzo, Ari Lerner, David Guttman, and Nate Murray

"React Design Patterns and Best Practices: Build easy to scale modular applications using the most powerful components and design patterns" by Carlos Santana Roldán

Course Contents for Discipline Specific Electives – DSE (Elective Group Basket)

Elective Group	**Elective 1** CCS/DSET/520-524	**Elective 2** CCS/DSET/570-574	**Elective 3** CCS/DSET/620-624	**Elective 4** CCS/DSET/670-674
Pattern Analysis & Machine Intelligence	Soft Computing	Fuzzy Systems : Theory, Application & Case Study	Video Processing	Pattern Recognition
Remote Sensing and Geospatial Technology	Fundamental of Satellite Remote Sensing	GIS	Remote Sensing And Digital Image Analysis	Hyperspectral Image Analysis
Security	Network Security	Cyber Security	Cyber Forensics: Tools, Techniques and Case Studies	Cryptography & Blockchain
Natural Language Processing	Linguistic Fundamentals: Understanding Language Structure and Analysis	Semantics and Pragmatics	Natural Language Processing	AI Chatbot Services and Applications

Note: *,# Student is advised to select the any one course from the pool of courses, however horizontal selection of courses to be followed at the time of selection of the course.

Elective Group – Pattern Analysis and Machine Intelligence

Course Type(**Mandatory)**

Course Code	Course Title	Credits	Contact Hours
CCS/DSET/520	**Soft Computing**	2	**2hrs/per week**
CCS/DSEP/520	Practical Based on Soft Computing	2	**4hrs/per week**

Course Description:

This course on soft computing aims to provide students with a comprehensive understanding of the fundamental concepts and applications of soft computing techniques. Soft computing encompasses a range of methodologies that enable computers to deal with uncertain, imprecise, and incomplete information. The course will cover various aspects of soft computing, including neural networks, fuzzy logic, and evolutionary computation, along with their practical applications. The course will incorporate case studies and reference books to enhance the students' understanding and application of soft computing techniques.

Prerequisites:

- Basic knowledge of mathematics and probability theory
- Understanding of computer programming concepts
- Familiarity with algorithms and data structures
- Basic understanding of artificial intelligence and machine learning

Course Objectives (CO):

- To introduce students to the fundamental concepts and techniques of soft computing
- To provide a comprehensive understanding of fuzzy logic systems, neural networks, and evolutionary computation
- To explore the integration of different soft computing methodologies
- To equip students with the skills to apply soft computing techniques to real-world problems
- To analyze and implement case studies that demonstrate the effectiveness of soft computing approaches

Learning Outcome (LO):

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

- Understand the principles and theories underlying soft computing techniques
- Apply fuzzy logic systems, neural networks, and evolutionary computation to solve complex problems
- Analyze and evaluate the performance of soft computing models
- Design and develop soft computing-based solutions for real-world case studies
- Critically assess the suitability of soft computing approaches in different domains

Course Outline:

Unit 1: Introduction to Soft Computing - Introduction to soft computing and its significance, Comparison with traditional computing paradigms, Components of soft computing: Neural networks, fuzzy logic, and evolutionary computation, Soft computing in real-world applications. Neural Networks - Introduction to artificial neural networks (ANNs), Single-layer perceptron and multi-layer perceptron, Training algorithms: Backpropagation, gradient descent, and variants, Deep learning and convolutional neural networks (CNNs), Case study: Image recognition using ANNs

Unit 2: Fuzzy Logic - Introduction to fuzzy logic and fuzzy sets, Fuzzy rules and linguistic variables, Fuzzy inference systems and rule-based reasoning, Fuzzy control systems and applications Case study: Fuzzy logic-based temperature control system. Evolutionary Computation - Introduction to evolutionary computation and genetic algorithms, Representation schemes and fitness evaluation, Selection, crossover, and mutation operators, Genetic programming and applications Case study: Optimization using genetic algorithms

Unit 3: Hybrid Approaches and Applications - Hybridization of soft computing techniques, Neuro-fuzzy systems and their applications, Evolutionary neural networks, Soft computing in data mining, pattern recognition, and decision support systems, Case study: Hybrid approach for stock market prediction

Recommended Reference Books:

"Soft Computing: Techniques and Applications" by S. N. Sivanandam and S. N. Deepa

"Neural Networks and Learning Machines" by Simon Haykin

"Fuzzy Logic with Engineering Applications" by Timothy J. Ross

"Introduction to Evolutionary Computing" by A. E. Eiben and J. E. Smith

"Hybrid Intelligent Systems: Analysis and Design" by Siddhartha Bhattacharyya and Paramartha Dutta

Course Type(**Mandatory)**			
Course Code	**Course Title**	**Credits**	**Contact Hours**
CCS/DSET/570	**Fuzzy Systems : Theory, Application and Case Studies**	2	**2hrs/per week**
CCS/DSEP/570	Practical Based on Fuzzy Systems : Theory, Application and Case Studies	2	**4hrs/per week**

Course Description:

This course on Fuzzy Systems provides students with a comprehensive understanding of the theory, applications, and case studies related to fuzzy logic and fuzzy sets. Fuzzy logic is a powerful computational tool for handling uncertainty and imprecision in decision-making processes. The course will cover the fundamental concepts of fuzzy sets, fuzzy logic, and fuzzy reasoning, as well as explore their applications in various fields such as engineering, finance, and medicine. Through a combination of lectures, case studies, and practical exercises, students will develop the necessary skills to apply fuzzy systems in real-world scenarios.

Prerequisites:

- Basic knowledge of mathematics, including set theory and calculus.
- Familiarity with basic concepts of logic and reasoning.
- Basic programming skills (preferably in a language such as Python or MATLAB).

Course Objectives (CO):

- Understand the theoretical foundations of fuzzy systems and their applications.
- Gain knowledge of fuzzy logic, fuzzy sets, and fuzzy inference systems.
- Learn techniques for fuzzy system modeling, control, and decision-making.
- Develop the ability to design and implement fuzzy systems for real-world problems.
- Analyze and evaluate case studies to understand the practical implications of fuzzy systems.

Learning Outcomes (LO):

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

- Explain the principles and concepts underlying fuzzy systems.
- Design and implement fuzzy inference systems for various applications.
- Apply fuzzy modeling techniques to develop solutions for complex problems.
- Evaluate and analyze the performance of fuzzy systems in real-world scenarios.
- Apply fuzzy decision-making methods to support complex decision processes.

Course Outline:

Unit 1: Introduction to Fuzzy Systems - Introduction to uncertainty and imprecision, Fuzzy sets and membership functions, Fuzzy operations and linguistic variables, Fuzzy rules and fuzzy reasoning. Fuzzy Systems Design and Implementation - Fuzzy inference systems, Rule-based fuzzy systems, Fuzzy control systems, Fuzzy decision-making

Unit 2: Fuzzy Systems in Engineering Applications - Fuzzy control systems in robotics, Fuzzy modelling and control in process industries, Fuzzy systems in intelligent transportation systems, Fuzzy systems for pattern recognition. Fuzzy Systems in Finance and Economics - Fuzzy logic in portfolio optimization, Fuzzy modeling for risk assessment, Fuzzy time series forecasting, Fuzzy systems in credit scoring

Unit 3: Fuzzy Systems in Medical and Healthcare Applications - Fuzzy expert systems for medical diagnosis, Fuzzy systems for healthcare resource allocation, Fuzzy decision support systems in medical treatment, Fuzzy modeling of patient satisfaction, Case Studies a) Analyzing and implementing fuzzy systems using software tools, b) Case studies from various domains (engineering, finance, medicine), c) Evaluation and performance assessment of fuzzy systems

Recommended Reference Books:

"Fuzzy Logic with Engineering Applications" by Timothy J. Ross

"Fuzzy Sets and Fuzzy Logic: Theory and Applications" by George J. Klir and Bo Yuan

"Introduction to Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems" by Guanrong Chen and Trung Tat Pham

"Fuzzy Systems Engineering: Theory and Practice" by C. L. Philip Chen and Han-Pang Huang

"Fuzzy Logic: Intelligence, Control, and Information" by John Yen and Reza Langari

Course Type(**Mandatory)**			
Course Code	**Course Title**	**Credits**	**Contact Hours**
CCS/DSET/620	**Video Processing**	2	**2hrs/per week**
CCS/DSEP/620	Practical Based on Video Processing	2	**4hrs/per week**

Course Description:

Video Processing with Python: Case Studies is an advanced course that explores the principles and techniques of video processing using the Python programming language. This course focuses on practical applications and provides hands-on experience with real-world case studies. Students will learn how to manipulate, analyze, and enhance video data, enabling them to extract meaningful information and insights from video streams.

Prerequisites:

- Proficiency in Python programming language
- Basic understanding of image processing concepts
- Familiarity with fundamental concepts of computer vision

Course Objectives (CO):

- Understand the fundamental principles and techniques of video processing.
- Gain proficiency in using Python libraries and frameworks for video manipulation and analysis.
- Apply various video processing techniques to extract meaningful information from video streams.
- Acquire the skills to enhance and restore video quality using advanced algorithms.
- Explore the application of deep learning for video analysis and synthesis.
- Develop the ability to design and implement video processing solutions for real-world problems.

Learning Outcomes (LO):

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

- Understand the underlying concepts and techniques of video processing.
- Manipulate and preprocess video data using Python.
- Implement motion detection, object recognition, and face detection algorithms.
- Enhance video quality through denoising, deblurring, and color correction.
- Apply video compression techniques for efficient storage and transmission.
- Utilize deep learning models for video analysis, classification, and synthesis.
- Design and develop video processing solutions for real-world applications.

Course Outline:

Unit 1: Introduction to Video Processing - Fundamentals of video representation and formats, Video acquisition and preprocessing techniques. Video Analysis Techniques - Motion detection and tracking, Object recognition and tracking, Face detection and recognition

Unit 2: Video Enhancement and Restoration - Noise reduction and image stabilization, Video denoising and deblurring, Contrast enhancement and color correction. Video Compression and Encoding, Principles of video compression, Video encoding algorithms and standards, Codecs and formats for efficient storage and transmission

Unit 3: Deep Learning for Video Processing - Convolutional neural networks for video analysis, Video classification and action recognition, Video generation and synthesis. Case Studies a) Real-world applications of video processing using Python, b) Case studies on surveillance systems, video analytics, etc. c) Hands-on projects to reinforce concepts learned.

Recommended Reference Book:

"Python for Video Processing: Techniques and Case Studies"

"Digital Video Processing", Second Edition by A. Murat Tekalp, June 2015, Pearson, ISBN: 9780133991116

Course Type(**Mandatory)**			
Course Code	**Course Title**	**Credits**	**Contact Hours**
CCS/DSET/670	**Pattern Recognition**	2	**2hrs/per week**
CCS/DSEP/670	Practical Based on Pattern Recognition	2	**4hrs/per week**

Course Description:
This course is designed to provide students with a comprehensive understanding of pattern classification using Python. Pattern classification is a fundamental technique in machine learning and data analysis, and this course aims to equip students with the necessary knowledge and skills to effectively apply pattern classification algorithms to real-world problems. Through a combination of theoretical concepts, practical exercises, and case studies, students will gain hands-on experience in implementing pattern classification algorithms using Python.

Prerequisites:
- Basic knowledge of Python programming language
- Familiarity with fundamental concepts in machine learning
- Understanding of linear algebra and probability theory

Course Objectives (CO):
- To understand the fundamental principles and techniques of pattern classification
- To develop proficiency in implementing pattern classification algorithms using Python
- To gain practical experience through hands-on exercises and real-world case studies
- To learn how to evaluate and select appropriate classification models for different tasks
- To explore the ethical considerations and challenges in pattern classification

Learning Outcome (LO):
By the end of this course, students will be able to:
- Understand the concepts and principles of pattern classification
- Apply various pattern classification algorithms using Python
- Preprocess and analyze datasets for pattern classification tasks
- Evaluate and compare the performance of classification models
- Implement pattern classification solutions to real-world problems

Course Outline:

Unit 1: Introduction to Pattern Classification - Overview of pattern classification, Importance and applications of pattern classification, Key concepts and terminology. Data Preprocessing and Feature Selection - Data cleaning and preprocessing techniques, Feature extraction and dimensionality reduction methods, Feature selection algorithms

Unit 2: Supervised Learning Algorithms - Linear classifiers (e.g., logistic regression, support vector machines), Decision trees and ensemble methods (e.g., random forests, boosting), Neural networks and deep learning models. Unsupervised Learning Algorithms - Clustering techniques (e.g., k-means, hierarchical clustering), Density estimation and Gaussian mixture models, Dimensionality reduction methods (e.g., principal component analysis)

Unit 3: Evaluation Metrics and Model Selection - Performance evaluation measures (e.g., accuracy, precision, recall), Cross-validation techniques, Model selection and hyperparameter tuning. Pattern Classification with Python - Overview of Python libraries for pattern classification (e.g., scikit-learn, TensorFlow, Keras), Implementing classification algorithms using Python, Handling large datasets and optimizing performance.

Recommended Reference Book:
"Pattern Classification and Machine Learning", Christopher M. Bishop, Springer, 2006
"Pattern Classification", Richard O. Duda, Peter E. Hart, David G. Stork, Wiley

Elective Group – Remote Sensing and Geo Spatial Technology

Course Type(**Mandatory)**			
Course Code	**Course Title**	**Credits**	**Contact Hours**
CCS/DSET/521	**Soft Computing**	2	**2hrs/per week**
CCS/DSEP/521	Practical Based on Soft Computing	2	**4hrs/per week**

Course Description:

This course provides a comprehensive introduction to the field of satellite remote sensing, focusing on the principles, techniques, and applications of using satellite-based sensors to collect and analyze Earth observation data. Students will gain a solid understanding of the fundamental concepts and tools involved in satellite remote sensing and explore various case studies to illustrate the practical applications of this technology in different domains.

Prerequisite:

Students should have a basic understanding of geographic information systems (GIS) and remote sensing principles. Familiarity with computer programming and image processing software would be advantageous but not mandatory.

Course Objectives (CO):

- To introduce students to the principles and techniques of satellite remote sensing
- To develop students' skills in analyzing and interpreting satellite imagery
- To provide hands-on experience with remote sensing software and tools
- To explore the applications of satellite remote sensing in various domains
- To enhance students' ability to critically evaluate and utilize satellite data for real-world case studies

Learning Outcomes (LO):

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

- Understand the fundamental concepts and principles of satellite remote sensing
- Acquire, preprocess, and enhance satellite imagery for analysis

- Perform image classification and interpretation using appropriate techniques
- Apply satellite remote sensing for land cover mapping and change detection
- Apply satellite remote sensing for specific applications, such as agriculture, environment, and disaster assessment
- Critically analyze and evaluate remote sensing data for case studies

Course Outline:

Unit 1: Introduction to Remote Sensing - Overview of remote sensing principles and technologies, Types of remote sensing platforms, Electromagnetic spectrum and its interaction with Earth's surface, Satellite Systems and Sensors - Overview of satellite systems and orbits, Types of satellite sensors and their characteristics, Sensor resolution and image interpretation

Unit 2: Image Acquisition and Preprocessing - Geometric and radiometric corrections, Image enhancement techniques, Image fusion and multi-temporal analysis. Image Classification and Interpretation - Supervised and unsupervised classification methods, Object-based image analysis, Land cover and land use mapping

Unit 3: Digital Elevation Models (DEMs) and Topographic Analysis - Generation and applications of DEMs, Terrain modeling and slope analysis, Hydrological applications. Change Detection and Time Series Analysis - Techniques for detecting and monitoring changes, Analysis of temporal satellite data, Urban growth and deforestation studies. Case Studies & Applications of Satellite Remote Sensing a) Agriculture and crop monitoring, b) Environmental monitoring and natural resource management, c) Disaster assessment and mitigation, d) Urban planning and infrastructure development

Recommended Reference Book:

"Remote Sensing of the Environment: An Earth Resource Perspective" by John R. Jensen

"Remote Sensing and Image Interpretation" by Thomas Lillesand, Ralph W. Kiefer, and Jonathan Chipman

"Introduction to Satellite Remote Sensing: Atmosphere, Ocean, Land and Cryosphere Applications" by William Emery, Sr. and Joel Susskind

Course Type(**Mandatory)**			
Course Code	**Course Title**	**Credits**	**Contact Hours**
CCS/DSET/571	**Geographic Information System (GIS)**	2	**2hrs/per week**
CCS/DSEP/571	Practical Based on Geographic Information System (GIS)	2	**4hrs/per week**

Course Description:

This GIS course is designed to provide students with an in-depth understanding of advanced concepts, techniques, and applications in Geographic Information Systems (GIS). The course focuses on enhancing students' knowledge and skills in spatial analysis, data management, cartography, and geospatial modeling. Through a combination of theoretical knowledge and practical hands-on exercises, students will gain the expertise required to address complex spatial problems and develop solutions using advanced GIS tools and methodologies.

Prerequisite:

- Basic knowledge of Geographic Information Systems (GIS)
- Familiarity with spatial data concepts and formats
- Proficiency in using GIS software (e.g., ArcGIS, QGIS)

Course Objectives (CO):

- Develop a deep understanding of advanced GIS concepts and techniques
- Acquire practical skills in spatial analysis and modeling
- Enhance proficiency in data management and integration
- Explore advanced cartography and visualization techniques
- Gain knowledge in geospatial modeling and simulation
- Apply advanced GIS methods to solve real-world problems

Learning Outcomes (LO):

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

- Apply advanced spatial analysis techniques to solve complex spatial problems
- Design and implement advanced data management strategies in GIS
- Create visually compelling and informative maps using advanced cartographic techniques
- Develop geospatial models and simulations to understand and predict spatial phenomena
- Evaluate and select appropriate GIS tools and methodologies for specific applications
- Apply critical thinking skills to analyze and solve real-world spatial problems using GIS

Course Outline:

Unit 1: Introduction to GIS Concepts - Overview of advanced GIS techniques and applications, Spatial analysis and modelling, Data integration and interoperability, Advanced Spatial Analysis Techniques - Spatial statistics and analysis, Network analysis and routing, Geo-statistics and interpolation, Multicriteria decision analysis (MCDA), Spatial regression analysis

Unit 2: Advanced Data Management in GIS - Database design and management, Data quality assessment and improvement, Data integration and fusion, Spatial data infrastructure (SDI) concepts, Data privacy and security in GIS

Unit 3: Advanced Cartography and Visualization - Advanced cartographic design principles, Thematic mapping techniques, 3D visualization and virtual reality (VR) in GIS, Web-based mapping and interactive applications, Data-driven cartography and dynamic mapping, Geospatial Modelling and Simulation - Introduction to geospatial modelling, Cellular automata and agent-based modelling, Time-series analysis and forecasting, Risk assessment and hazard modelling, Urban growth modelling

Recommended Reference Book:

"Advanced Geographic Information Systems: Spatial Analysis and Modeling", Robert P. Haining, Wiley, 2021

Course Type(**Mandatory)**			
Course Code	**Course Title**	**Credits**	**Contact Hours**
CCS/DSET/521	**Digital Remote Sensing Image Analysis**	2	**2hrs/per week**
CCS/DSEP/521	Practical Based on Digital Remote Sensing Image Analysis	2	**4hrs/per week**

Course Description:

This course provides an in-depth exploration of digital remote sensing image analysis techniques, with a particular focus on the analysis of satellite and aerial imagery. Students will gain a comprehensive understanding of the principles, methodologies, and practical applications of remote sensing image analysis. The course will cover various case studies to illustrate the real-world applications of digital remote sensing in different fields.

Prerequisite:

- Basic knowledge of geography, environmental science, or a related field.
- Familiarity with basic computer skills and image processing concepts.

Course Objectives (CO):

- To provide students with a solid understanding of remote sensing principles and technologies.
- To develop practical skills in digital image processing and analysis techniques.
- To explore the various applications of remote sensing in different domains.
- To enhance critical thinking and problem-solving abilities through case study analysis.
- To prepare students for careers in remote sensing, geospatial analysis, and related fields.

Learning Outcomes (LO):

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

- Understand the fundamental principles and technologies of remote sensing.
- Apply image pre-processing techniques to enhance and correct remote sensing data.
- Perform image classification, segmentation, and change detection using remote sensing imagery.
- Analyze and interpret remote sensing data for different applications and domains.
- Effectively use remote sensing software and tools for image analysis.
- Demonstrate critical thinking skills by evaluating and discussing case studies.

Course Outline:

Unit 1: Introduction to Remote Sensing - Overview of remote sensing principles and technologies, Types of remote sensing platforms and sensors, Introduction to satellite and aerial imagery, Image Pre-processing Techniques - Image acquisition and data formats, Radiometric and geometric correction methods, Atmospheric correction techniques, Image enhancement and noise reduction

Unit 2: Image Classification and Segmentation - Supervised and unsupervised classification methods, Feature extraction techniques, Object-based image analysis (OBIA), Change detection and monitoring - Change detection techniques, Temporal analysis of remote sensing data, Monitoring land cover changes. Hyperspectral Image Analysis: Hyperspectral data characteristics, Spectral unmixing, Classification of hyperspectral images. Case Studies and Applications: Land cover mapping, Vegetation monitoring, Urban growth analysis, Environmental monitoring

Unit 3: Advanced Image Analysis Techniques - Hyperspectral and multispectral image analysis, Fusion of remote sensing data sources, Texture analysis and spatial pattern recognition, Time-series analysis, Applications of Digital Remote Sensing, Land cover and land use mapping, Environmental monitoring and assessment, Urban planning and infrastructure management, Agriculture and forestry applications, Disaster management and emergency response.

Recommended Reference Book:

"Digital Image Processing: Remote Sensing Perspectives", John R. Jensen, Prentice Hall, 2016

"Remote Sensing and Image Interpretation" Thomas M. Lillesand, Ralph W. Kiefer, and Jonathan W. Chipman Wiley

Course Type(**Mandatory)**			
Course Code	**Course Title**	**Credits**	**Contact Hours**
CCS/DSET/671	**Hyper Spectral Image Analysis**	2	**2hrs/per week**
CCS/DSEP/671	Practical Based on Hyper Spectral Image Analysis	2	**4hrs/per week**

Course Description:

This Hyperspectral Image Analysis course is designed to provide students with an in-depth understanding of hyperspectral imaging techniques and their applications in various fields. The course will focus on advanced methods and algorithms used for processing and analyzing hyperspectral data, including case studies that showcase real-world applications. Students will gain practical skills in handling hyperspectral data, extracting meaningful information, and solving complex problems related to spectral image analysis.

Prerequisites:

- Basic knowledge of remote sensing principles and techniques
- Familiarity with image processing fundamentals
- Understanding of statistical concepts

Course Objectives (CO):

- Gain a comprehensive understanding of hyperspectral imaging principles and techniques
- Develop proficiency in preprocessing hyperspectral data for analysis
- Learn advanced algorithms and methods for hyperspectral image classification and unmixing
- Acquire skills in dimensionality reduction and feature extraction from hyperspectral data

- Explore advanced analysis techniques and their applications in real-world case studies

Learning Outcomes (LO):

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

- Describe the principles of hyperspectral imaging and its applications
- Perform preprocessing tasks such as radiometric correction and noise reduction
- Apply supervised and unsupervised classification techniques to hyperspectral data
- Conduct endmember extraction and abundance estimation for unmixing
- Implement dimensionality reduction methods to reduce data complexity
- Apply advanced analysis techniques for target detection, sub-pixel mapping, and change detection
- Analyze and interpret hyperspectral data in the context of various case studies

Course Outline:

Unit 1: Introduction to Hyperspectral Imaging - Basics of hyperspectral data acquisition and characteristics, Spectral signatures and spectral libraries. Hyperspectral Data Preprocessing - Radiometric and atmospheric correction, Noise reduction techniques, Calibration and geometric correction

Unit 2: Hyperspectral Image Classification - Supervised and unsupervised classification methods, Feature extraction and selection, Neural networks and deep learning for classification, Hyperspectral Unmixing - Linear and nonlinear unmixing algorithms, Endmember extraction, Abundance estimation and abundance maps

Unit 3: Dimensionality Reduction - Principal Component Analysis (PCA), Independent Component Analysis (ICA), Non-negative Matrix Factorization (NMF). Advanced Hyperspectral Analysis Techniques - Target detection and anomaly detection, Sub-pixel mapping, Change detection and time-series analysis, Case Studies in Hyperspectral Image Analysis – a) Applications in agriculture, b) environmental monitoring, c) mineral exploration, and remote sensing

Recommended Reference Book:

Hyperspectral Remote Sensing: Principles and Applications, Antonio Plaza, Jon Atli Benediktsson, Josiane Zerubia, Academic Press, 2016

Elective Group – Security

Course Type(**Mandatory)**			
Course Code	**Course Title**	**Credits**	**Contact Hours**
CCS/DSET/522	**Network Security**	2	**2hrs/per week**
CCS/DSEP/522	Practical Based on Network Security	2	**4hrs/per week**

Course Description:

This course provides a comprehensive introduction to network security principles and practices. Students will learn about various threats, vulnerabilities, and attack vectors in network environments, as well as the methods and technologies used to secure networks. The course will also include case studies and real-world examples to illustrate the practical application of network security concepts. Students will have the opportunity to explore reference books and resources to deepen their understanding of the subject matter.

Prerequisites:

- Basic knowledge of networking concepts
- Familiarity with operating systems and computer architecture
- Understanding of fundamental security principles

Course Objectives (CO):

- Understand the fundamental principles and concepts of network security.

- Identify and assess network security risks and vulnerabilities.
- Evaluate and implement appropriate network security technologies and protocols.
- Develop incident response plans and perform network forensics.
- Analyze and apply security measures in cloud and mobile network environments.
- Analyze and learn from real-world network security case studies.

Learning Outcomes (LO):

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

- Apply network security principles to design secure network infrastructures.
- Implement and configure network security technologies effectively.
- Analyze and respond to network security incidents.
- Evaluate and apply security measures in cloud and mobile networks.
- Investigate and present case studies of network security breaches.

Course Outline:

Unit 1: Introduction to Network Security - Overview of network security concepts, Security goals: confidentiality, integrity, availability, Threat landscape and attack vectors, Risk assessment and management, Network Architecture and Design - Secure network design principles, Segmentation and zoning, Defense-in-depth strategies, Security considerations for wireless and mobile networks

Unit 2: Network Perimeter Security - Firewalls and intrusion detection/prevention systems, Virtual Private Networks (VPNs), Network Address Translation (NAT), Demilitarized Zone (DMZ) design, Secure Network Protocols and Services - Secure Socket Layer/Transport Layer Security (SSL/TLS), Secure Shell (SSH), Domain Name System Security Extensions (DNSSEC), Network Time Protocol Security (NTPsec)

Unit 3: Intrusion Detection and Prevention Systems (IDPS) - Host-based and network-based IDPS, Signature-based and anomaly-based detection, Incident response and handling. Wireless Network Security - Wi-Fi security protocols (WEP, WPA, WPA2, WPA3), Rogue access point detection, Wireless intrusion prevention systems (WIPS), Securing mobile devices and applications. Case Studies in Network Security a) Analysis of real-world security breaches b) Investigating network attacks and incidents c) Lessons learned from notable security incidents

Recommended Reference Books:

"Network Security: Private Communication in a Public World" by Charlie Kaufman, Radia Perlman, and Mike Speciner

"Firewalls and Internet Security: Repelling the Wily Hacker" by William R. Cheswick and Steven M. Bellovin

"Network Security Essentials: Applications and Standards" by William Stallings

"Practical Packet Analysis: Using Wireshark to Solve Real-World Network Problems" by Chris Sanders

"The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws" by Dafydd Stuttard and Marcus Pinto

"Hacking: The Art of Exploitation" by Jon Erickson

"Applied Cryptography: Protocols, Algorithms, and Source Code in C" by Bruce Schneier

"Security Engineering: A Guide to Building Dependable Distributed Systems" by Ross J. Anderson

Course Type(**Mandatory)**			
Course Code	**Course Title**	**Credits**	**Contact Hours**
CCS/DSET/572	**Cyber Security**	2	**2hrs/per week**
CCS/DSEP/572	Practical Based on Cyber Security	2	**4hrs/per week**

Course Description:

This course provides a comprehensive overview of cyber security, focusing on various tools, practical case studies, and essential reference books. Students will learn the fundamental concepts, techniques, and best practices in cyber security, gaining hands-on experience with popular tools used in the industry. The course incorporates real-world case studies to illustrate the application of these tools and concepts. Students will also explore relevant reference books to deepen their understanding and develop a strong foundation in cyber security.

Prerequisite:

Basic understanding of computer networks and information technology.

Course Objectives (CO):

- To provide students with a solid foundation in cyber security concepts and principles.
- To develop an understanding of common cyber threats and vulnerabilities.
- To familiarize students with various tools and techniques used in cyber security.
- To enable students to assess and mitigate risks to digital systems and data.
- To promote ethical considerations and legal compliance in cyber security practices.

Learning Outcomes (LO):

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

- Identify and analyze common cyber threats and vulnerabilities.
- Implement network security measures to protect against attacks.
- Apply cryptographic techniques to ensure data confidentiality and integrity.
- Develop and implement secure software development practices.
- Demonstrate incident response and management skills.
- Conduct ethical hacking and penetration testing activities.
- Evaluate and recommend security measures for different scenarios.

Course Outline:

Unit 1: Introduction to Cyber Security - Overview of cyber security fundamentals, Importance of cyber security in modern society, Legal and ethical considerations. Cyber Threats and Attacks - Types of cyber threats: malware, social engineering, phishing, etc. Attack vectors and methods, Risk assessment and vulnerability analysis

Unit 2: Cybersecurity Tools - Firewall and intrusion detection systems, Anti-malware software and endpoint protection, Network monitoring and log analysis tools, Encryption and data protection tools, Securing Networks and Systems - Network security principles and protocols, Secure configuration of operating systems and applications, Authentication and access control mechanisms, Patch management and vulnerability mitigation

Unit 3: Incident Response and Forensics - Incident response lifecycle, Digital forensics techniques, Evidence collection and analysis, Post-incident recovery and lessons learned, Case Studies in Cyber Security – a) Analysis of real-world cyber security incidents, b) Examination of incident response strategies c) Lessons learned and best practices

Recommended Reference Books:

"The Art of Invisibility" by Kevin Mitnick

"Hacking: The Art of Exploitation" by Jon Erickson

"Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software" by Michael Sikorski and Andrew Honig

"The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws" by Dafydd Stuttard and Marcus Pinto

"Network Security Bible" by Eric Cole

"Blue Team Handbook: Incident Response Edition" by Don Murdoch, Jr.

"Digital Forensics and Incident Response: Developing an Incident Response Plan" by Gerard Johansen

Course Type(**Mandatory)**			
Course Code	**Course Title**	**Credits**	**Contact Hours**
CCS/DSET/622	**Cyber Forensics: Tools, Techniques, and Case Studies**	2	**2hrs/per week**
CCS/DSEP/622	Practical Based on Cyber Forensics: Tools, Techniques, and Case Studies	2	**4hrs/per week**

Course Description:

This course is designed to provide students with a comprehensive understanding of the field of cyber forensics, including the tools and techniques used for investigating and analyzing digital evidence. The course will cover the fundamental concepts of cyber forensics, explore various forensic tools and methodologies, and present real-world case studies to demonstrate the application of these tools in solving cybercrime investigations.

Prerequisites:

- Basic knowledge of computer systems and networks
- Familiarity with operating systems and file systems
- Understanding of information security principles

Course Objectives (CO):

- Understand the fundamentals of cyber forensics and its role in combating cybercrime.
- Gain proficiency in the use of tools and techniques for acquiring and analyzing digital evidence.
- Develop skills in conducting forensic investigations and incident response procedures.
- Comprehend the legal and ethical issues associated with cyber forensics.
- Explore advanced topics such as network forensics, mobile and cloud forensics, and malware analysis.
- Apply theoretical knowledge to real-world case studies and practical scenarios.

Learning Outcome (LO):

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

- Identify and collect digital evidence using forensically sound methods.
- Analyze digital artifacts to uncover evidence of cybercrime.
- Conduct network and mobile device forensics investigations.
- Employ appropriate tools and techniques to recover and preserve data.
- Understand the legal and ethical considerations in cyber forensics.

Course Outline:

Unit 1: Introduction to Cyber Forensics - Understanding cyber forensics: scope and importance, Legal and ethical considerations in cyber investigations, Roles and responsibilities of a cyber forensic examiner. Digital Evidence and Forensic Processes, Types of digital evidence, Evidence acquisition and preservation, Forensic imaging and hashing, Chain of custody and documentation

Unit 2: Forensic Tools and Techniques - Forensic imaging tools (e.g., EnCase, FTK, Autopsy), File system analysis and recovery, Network forensics and log analysis, Memory forensics and analysis, Mobile and Cloud Forensics, Forensics for mobile devices (smartphones, tablets), Investigating cloud-based platforms and services, Extraction and analysis of mobile app data

Unit 3: Network Traffic Analysis - Packet capture and analysis, Intrusion detection and prevention systems, Network log analysis, Malware Analysis and Reverse Engineering - Introduction to malware analysis, Static and dynamic analysis techniques, Code decompilation and reverse engineering. Case Studies and Practical Application – a) Analyzing a cybercrime case from start to finish, b) Examining digital evidence and conducting forensic analysis, c) Reporting findings and presenting evidence in court

Recommended Reference Books:

"Digital Forensics and Cyber Crime: An Introduction" by Marjie T. Britz

"Computer Forensics: Investigating Data and Image Files" by EC-Council

"The Basics of Digital Forensics: The Primer for Getting Started in Digital Forensics" by John Sammons

"File System Forensic Analysis" by Brian Carrier

"Malware Forensics: Investigating and Analyzing Malicious Code" by Cameron H. Malin, et al.

"Practical Packet Analysis: Using Wireshark to Solve Real-World Network Problems" by Chris Sanders

Course Type(**Mandatory)**			
Course Code	**Course Title**	**Credits**	**Contact Hours**
CCS/DSET/672	**Cryptography and BlockChain**	2	**2hrs/per week**
CCS/DSEP/672	Practical Based on Cryptography and BlockChain	2	**4hrs/per week**

Course Description:

This introductory course provides students with a comprehensive understanding of cryptography and blockchain technologies. Students will learn the fundamental principles, techniques, and applications of cryptography and explore the underlying concepts and mechanisms behind blockchain technology. Through case studies and real-world examples, students will gain practical knowledge and insight into the use of these technologies in various industries. The course aims to equip students with the necessary foundation to understand and work with cryptography and blockchain in a professional context.

Prerequisite:

- Basic understanding of computer science concepts and programming
- Familiarity with data structures and algorithms
- Knowledge of networking and internet protocols

Course Objectives (CO):

- Understand the fundamental concepts and principles of cryptography and blockchain technology.
- Analyze and evaluate different cryptographic algorithms, protocols, and systems.
- Gain hands-on experience in implementing and securing cryptographic systems.
- Explore the inner workings of blockchain networks and consensus mechanisms.
- Develop the skills to design and deploy decentralized applications using smart contracts.
- Investigate real-world use cases of cryptography and blockchain in various industries.

Learning Outcome (LO):

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

- Demonstrate a comprehensive understanding of cryptographic algorithms and their applications.
- Evaluate the security of cryptographic systems and propose countermeasures.
- Analyze and design blockchain networks for specific use cases.
- Develop and deploy smart contracts for decentralized applications.
- Apply cryptography and blockchain principles to address real-world challenges.

Course Outline:

Unit 1: Introduction to Cryptography - Overview of cryptography and its historical significance, Symmetric and asymmetric encryption algorithms, Hash functions and digital signatures, Cryptographic protocols and applications. Cryptographic Techniques and Algorithms - Key management and distribution, Public key infrastructure (PKI), Cryptographic attacks and countermeasures, Secure communication protocols (SSL/TLS)

Unit 2: Introduction to Blockchain Technology - Evolution of blockchain and its applications, Distributed ledger technology and consensus mechanisms, Cryptocurrencies and smart contracts, Blockchain platforms and ecosystems, Blockchain Security and Privacy - Blockchain vulnerabilities and attack vectors, Privacy and anonymity in blockchain, Tokenization and non-fungible tokens (NFTs), Securing blockchain networks and transactions

Unit 3: Future Trends and Challenges - Quantum cryptography and post-quantum encryption, Scalability and interoperability in blockchain, Regulatory and legal considerations, Ethical implications of cryptography and blockchain technology, Case Studies in Cryptography and Blockchain – a) Real-world applications of cryptography in finance, healthcare, and government sectors, b) Success stories and challenges of blockchain adoption in industries such as supply chain, voting, and identity management.

Recommended Reference Books:

"Cryptography and Network Security: Principles and Practice" by William Stallings

"Mastering Bitcoin: Unlocking Digital Cryptocurrencies" by Andreas M. Antonopoulos

"Blockchain Basics: A Non-Technical Introduction in 25 Steps" by Daniel Drescher

"Blockchain Revolution: How the Technology Behind Bitcoin and Other Cryptocurrencies is Changing the World" by Don Tapscott and Alex Tapscott

"The Age of Cryptocurrency: How Bitcoin and Digital Money are Challenging the Global Economic Order" by Paul Vigna and Michael J. Casey

"Cryptocurrency: How Bitcoin and Digital Money are Challenging the Global Economic Order" by Abraham K. White

Elective Group – Natural Language Processing

Course Type(**Mandatory)**			
Course Code	**Course Title**	**Credits**	**Contact Hours**
CCS/DSET/523	**Linguistic Fundamentals: Understanding Language Structure and Analysis**	2	**2hrs/per week**
CCS/DSEP/523	Practical Based on Linguistic Fundamentals: Understanding Language Structure and Analysis	2	**4hrs/per week**

Course Description:

This course Linguistic Fundamentals provides students with a comprehensive introduction to the principles and theories underlying language structure and analysis. Through a combination of theoretical concepts, case studies, and practical examples, students will develop a solid foundation in linguistic analysis and gain insights into the intricacies of human language. The course will cover key areas such as phonetics, phonology, morphology, syntax, semantics, and pragmatics, along with relevant case studies to illustrate real-world applications. Students will also be introduced to influential reference books in the field to deepen their understanding of linguistic fundamentals.

Prerequisite:

None. This course is open to all students with an interest in linguistics. Prior knowledge of linguistics is not required.

Course Objectives (CO):

- To develop a comprehensive understanding of the fundamental components of language.
- To examine the interaction between linguistic elements and their impact on language structure.
- To cultivate analytical and critical thinking skills in linguistic analysis.
- To explore language variation and its social and cultural implications.
- To apply linguistic theories and methods to real-world case studies.

Learning Outcomes (LO):

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

- Identify and describe the core elements of language, including phonetics, phonology, morphology, syntax, semantics, and pragmatics.
- Analyze and transcribe speech sounds using phonetic symbols.
- Conduct morphological and syntactic analyses of linguistic data.
- Interpret and analyze meaning at the lexical, sentence, and discourse levels.
- Apply sociolinguistic principles to examine language variation and change.
- Apply linguistic theories and methods to case studies, demonstrating critical thinking skills.

Course Outline:

Unit 1: Introduction to Linguistic Fundamentals - Overview of linguistics as a field of study, Language as a system of communication, Fundamental branches of linguistics. Phonetics and Phonology - Introduction to phonetics: articulatory, acoustic, and auditory aspects, Phonemic analysis: phonemes, allophones, and phonological rules, Case study: Phonological variations across different dialects

Unit 2: Morphology - Basic units of meaning: morphemes, Word formation processes: affixation, compounding, derivation, etc. Case study: Analyzing morphological structures in different languages. Syntax - Sentence structure and phrase types, Syntactic categories and constituents, Case study: Parsing and analyzing sentence structures

Unit 3: Semantics and Pragmatics - Meaning and interpretation of words, phrases, and sentences, Pragmatic principles and implicatures, Case study: Pragmatic analysis of speech acts, Language and Society - Sociolinguistics: language variation and social factors, Language acquisition and bilingualism Case study: Language policy and planning

Recommended Reference Books:

"An Introduction to Language" by Victoria Fromkin, Robert Rodman, and Nina Hyams

"Linguistics: An Introduction to Linguistic Theory" by Victoria A. Fromkin, Robert Rodman, and Nina Hyams

"The Cambridge Encyclopedia of Language" by David Crystal

"Language Files: Materials for an Introduction to Language and Linguistics" by Department of Linguistics, Ohio State University

"Introducing Morphology" by Rochelle Lieber

Course Type(**Mandatory)**			
Course Code	**Course Title**	**Credits**	**Contact Hours**
CCS/DSET/573	**Semantics and Pragmatics**	2	**2hrs/per week**
CCS/DSEP/573	Practical Based on Semantics and Pragmatics	2	**4hrs/per week**

Course Description:

This course provides an in-depth exploration of the fields of semantics and pragmatics. Semantics focuses on the study of meaning in language, while pragmatics examines the use of language in context. Students will gain a comprehensive understanding of the theoretical foundations, methodologies, and practical applications of semantics and pragmatics through lectures, discussions, case studies, and hands-on exercises.

Prerequisites:

- Basic knowledge of linguistics or a related field is recommended but not required.
- Familiarity with introductory-level syntax and morphology is beneficial.

Course Objectives (CO):

- Understand the key concepts and theories of semantics and pragmatics.
- Develop analytical skills to identify and analyze different layers of meaning in language.
- Recognize and evaluate the role of context in shaping linguistic interpretation.
- Apply theoretical knowledge to real-world case studies and practical scenarios.

- Enhance critical thinking and problem-solving abilities within the field of linguistics.
- Develop effective communication skills through class discussions and presentations.

Learning Outcomes (LO):

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

- Demonstrate a comprehensive understanding of the core principles of semantics and pragmatics.
- Analyze and interpret the meaning of words, sentences, and discourse in different contexts.
- Evaluate and discuss the impact of pragmatic factors on language use and interpretation.
- Apply theoretical frameworks to analyze and solve problems related to semantics and pragmatics.
- Construct well-supported arguments and engage in academic discussions on language meaning.
- Employ critical thinking to assess and interpret linguistic phenomena in various domains.

Course Outline:

Unit 1: Introduction to Semantics and Pragmatics - Overview of semantics and pragmatics, Distinction between meaning and use, Theoretical frameworks in semantics and pragmatics. Semantic Analysis - Word meaning and lexical semantics, Sentence meaning and compositional semantics, Truth-conditional semantics

Unit 2: Pragmatic Analysis - Context and implicature, Speech acts and illocutionary force, Conversational implicature and inference, Reference and Presupposition - Reference and referring expressions, Presupposition and presuppositional analysis, Anaphora and deixis

Unit 3: Meaning and Society - Pragmatics of politeness and face-saving, Cross-cultural and intercultural pragmatics, Gender and language use. Case Studies and Applications a) Analyzing discourse and conversation b) Pragmatics in legal and political discourse, c) Semantic and pragmatic analysis in advertising

Recommended Reference Books:

"Semantics: A Coursebook" by James R. Hurford, Brendan Heasley, and Michael B. Smith

"Pragmatics and Discourse: A Resource Book for Students" by Joan Cutting

"Meaning and Relevance" by Deirdre Wilson and Dan Sperber

"Pragmatics" by Stephen C. Levinson

"Semantics: An Introduction to Meaning in Language" by Kate Kearns

"Pragmatics: An Introduction" by Yan Huang

Course Type(**Mandatory)**			
Course Code	**Course Title**	**Credits**	**Contact Hours**
CCS/DSET/623	**Natural Language Processing**	2	**2hrs/per week**
CCS/DSEP/623	Practical Based on Natural Language Processing	2	**4hrs/per week**

Course Description:

This Natural Language Processing (NLP) aims to provide students with a foundational understanding of NLP techniques, methodologies, and applications. Students will explore the fundamental concepts of NLP, learn about popular algorithms and models, and gain practical experience through hands-on case studies. The course will also cover relevant reference books to help students deepen their understanding of NLP.

Prerequisites:

- Basic knowledge of programming concepts
- Familiarity with data structures and algorithms
- Understanding of probability and statistics
- Some exposure to machine learning concepts is beneficial but not mandatory

Course Objectives (CO):

- Understand the fundamental concepts and techniques in Natural Language Processing
- Develop practical skills to build and evaluate NLP systems
- Gain knowledge of various NLP applications and case studies
- Acquire the ability to analyze and solve NLP problems effectively
- Explore the challenges and ethical considerations in NLP

Learning Outcomes (LO):

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

- Explain the key concepts and methodologies in NLP
- Apply NLP techniques to preprocess and analyze text data
- Build and evaluate NLP models for tasks like sentiment analysis, named entity recognition, machine translation, and question answering
- Critically analyze and compare different approaches and algorithms in NLP
- Identify and address ethical concerns in NLP applications

Unit 1: Introduction to Natural Language Processing - Overview of NLP: Definition, history, and key applications. Language representation: Text preprocessing, tokenization, stemming, and lemmatization. Language modeling: N-grams, language models, and text generation. Evaluation metrics: Precision, recall, F1 score, and perplexity.

Unit 2: NLP Techniques and Algorithms - Text classification: Naive Bayes, logistic regression, and support vector machines (SVM), Sentiment analysis: Lexicon-based approaches, machine learning models, and deep learning models. Named Entity Recognition (NER): Rule-based systems, conditional random fields (CRF), and neural networks. Sequence labeling: Hidden Markov models (HMM) and Conditional Random Fields (CRF). Word embeddings: Word2Vec, GloVe, and FastText.

Unit 3: Advanced NLP - Topic modeling: Latent Dirichlet Allocation (LDA) and Non-negative Matrix Factorization (NMF). Text summarization: Extractive and abstractive approaches. Machine translation: Statistical methods, neural machine translation (NMT), and transformer models. Question answering: Information retrieval, document retrieval, and passage ranking. NLP in industry: Case studies of NLP applications in various domains, such as healthcare, finance, and customer support.

Recommended Reference Books:

"Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition" by Daniel Jurafsky and James H. Martin.

"Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit" by Steven Bird, Ewan Klein, and Edward Loper.

"Foundations of Statistical Natural Language Processing" by Christopher D. Manning and Hinrich Schütze.

"Deep Learning for Natural Language Processing" by Palash Goyal, Sumit Pandey, Karan Jain, and Karan Kumar.

"Text Mining and Analysis: Practical Methods, Examples, and Case Studies Using SAS" by Goutam Chakraborty and Murali Pagolu.

Course Type(**Mandatory)**			
Course Code	**Course Title**	**Credits**	**Contact Hours**
CCS/DSET/673	**AI Chatbot Services and Applications**	2	**2hrs/per week**
CCS/DSEP/673	Practical Based on AI Chatbot Services and Applications	2	**4hrs/per week**

Course Description:

This course provides an overview of AI chatbot services and their applications across various industries. Students will gain a solid understanding of the underlying technologies and concepts behind chatbot development and explore real-world case studies highlighting successful implementations. The course will also cover ethical considerations, best practices, and future trends in the field of AI chatbot services.

Prerequisites:

- Basic understanding of programming concepts
- Familiarity with Python or another programming language
- Knowledge of fundamental machine learning concepts

Course Objectives (CO):

- Understand the principles and technologies behind AI chatbot services.
- Gain hands-on experience in designing and developing chatbot systems.
- Explore the applications and potential use cases of chatbot technology.
- Acquire knowledge of natural language processing (NLP) techniques for chatbots.
- Develop skills in training and deploying chatbot models using machine learning.

Learning Outcome (LO):

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

- Design and implement functional chatbot systems using AI techniques.
- Apply NLP methods to process and interpret user input in chatbot interactions.
- Utilize machine learning algorithms to train and improve chatbot models.
- Evaluate and choose appropriate chatbot platforms and tools for specific applications.
- Consider ethical, legal, and privacy implications in chatbot development

Course Outline:

Unit 1: Introduction to AI Chatbot Services - Overview of AI chatbots and their significance, Historical development of chatbots, Applications and benefits of AI chatbot services, Introduction to natural language processing (NLP) and machine learning (ML) in chatbot development. Chatbot Design and Architecture - Chatbot design principles and user experience considerations, Conversational flow and dialogue management, Backend architecture and integration with existing systems, Introduction to chatbot development platforms and tools

Unit 2: Natural Language Processing (NLP) Fundamentals - Introduction to NLP and its role in chatbot services, Text preprocessing and tokenization, Named entity recognition (NER) and sentiment analysis, Word embeddings and language models, Machine Learning for Chatbot Development - Introduction to machine learning algorithms in chatbot development, Supervised, unsupervised, and reinforcement learning techniques, Training data collection and annotation, Evaluation metrics for chatbot performance

Unit 3: Building Rule-Based Chatbots - Rule-based chatbot development approach, Designing rule-based chatbot architectures, Implementing intent recognition and entity extraction using rule-based methods, Limitations and challenges of rule-based chatbots. Introduction to Conversational AI - Understanding conversational AI and its components, Introduction to intent recognition and dialogue management systems, Dialogflow and other conversational AI platforms, Creating intents, entities, and dialogues in conversational AI platforms. ek 10: Advanced Chatbot Features and Techniques - Multilingual chatbot development, Emotion detection and sentiment analysis in chatbots, Contextual understanding and maintaining conversation context, Implementing voice-based chatbot interfaces

Recommended Reference Books:

"Chatbots : An Introduction and Easy Guide to Building Your Own" by Matthew Harper

"Designing Bots: Creating Conversational Experiences" by Amir Shevat

"Building Chatbots with Python: Using Natural Language Processing and Machine Learning" by Sumit Raj

"Conversational AI and Chatbots: Fundamentals, Applications, and New Trends" by Manoj Prasad and Amitava Dasgupta

"Applied Artificial Intelligence: A Handbook for Business Leaders" by Mariya Yao, Adelyn Zhou, and Marlene Jia