VIDYASAGAR UNIVERSITY

Midnapore, West Bengal

PROPOSED CURRICULUM&SYLLABUS (DRAFT) OF

B. Sc. (HONOURS/ HONOURS WITH RESEARCH)

MAJOR IN COMPUTER SCIENCE

4-YEAR UNDERGRADUATE PROGRAMME

(w.e.f. Academic Year 2023-2024)

Based on

Curriculum & Credit Framework for Undergraduate Programmes

(CCFUP), 2023& NEP, 2020

VIDYASAGAR UNIVERSITY
BACHELOR OF SCIENCE (HONOURS/ HONOURS WITH RESEARCH) MAJOR IN COMPUTER SCIENCE
(under CCFUP, 2023)

Level	YR.	SEM	Course Type	Course Code	Course Title		Credit	L-T-P	Marks		
									CA	ESE	TOTAL
B.Sc. (Hons./ Hons. with Research)	4th	VII	SEMESTER-VII								
			Major-14	COSHMJ14	T: Software Engineering; P: Practical		4	3-0-1	15	60	75
			Major-15	COSHMJ15	T: Compiler Design; P: Practical		4	3-0-1	15	60	75
			CRM	COSHCRM	T: Research Methodology and Ethics;		4	3-1-0	15	60	75
			Major Elective-03	COSHDSE 03	Image Processing/ Big-Data/ Cyber-Security; P: Practical		4	3-0-1	15	60	75
			Minor-07 (Disc.-I)	COSMIN07	T: Digital Marketing *(To be taken by the other Discipline)*		4	3-1-0	15	60	75
						Semester-VII Total	20				375
		VIII	SEMESTER-VIII								
			Major-16	COSHMJ16	T: Artificial Intelligence; P: Practical		4	3-0-1	15	60	75
			Major Elective-04	COSHDSE04	T: Machine Learning/ Natural Language Processing/Data Science; P: Practical		4	3-0-1	15	60	75
			Major Elective-05	COSHDSE05	T: Advanced DBMS/Green Computing/ Cryptography and Steganography; P: Practical	FOR HONOURS STUDENTS	4	3-0-1	15	60	75
			Major Elective-06	COSHDSE06	T: Soft Computing/Quantum Computing/ Real Time System; P: Practical		4	3-0-1	15	60	75
			Research	COSHRD	DISSERTATION: Project / Research dissertation	*FOR HONOURS WITH RESEARCH STUDENTS*	*8*	*0-0-8*	*-*	*150*	*150*
			Minor-8 (Disc.-II)	COSMIN08	T; Introduction to Cloud Computing; P: Practical *(To be taken by the other Discipline)*		4	3-0-1	15	60	75
						Semester-VIII Total	20				375
						YEAR-4	40				750
			To be awarded *B.Sc. (Honours / Honours with Research) Major in Computer Science* on Completion				166	**Marks (Year: I+II+III+IV)**			3075

MJ = Major, MI = Minor Course, DSE = Discipline Specific Elective Course, CRM= Compulsory Research Methodology, CA= Continuous Assessment, ESE= End Semester Examination, T = Theory, P= Practical, L-T-P = Lecture-Tutorial-Practical

SEMESTER-VII

MAJOR (MJ)

MJ-14: Software Engineering **Credits 04 (Full Marks:75)**

MJ-14T: Software Engineering (Theory) *Credits 03 (45 hrs.)*

Course Outcome (CO):

Students able to:

CO1: Learn software engineering process models and metrics to plan and manage software development effectively.

CO2: Learn coding best practices, inspections, and walkthroughs to produce maintainable and reliable code.

CO3: Learn and execute effective test strategies using white-box, black-box, and specialized testing techniques.

CO4: Manage software maintenance processes, including cost estimation and reverse engineering, for long-term sustainability.

Course Content:

Unit 1: Introduction:

Software, Software Myths, Software types, Software engineering: A Layered Technology, Different Software Process Models, Fourth Generation Techniques, Software project management. [6L]

Unit 2: Software Design:

Problem Partitioning, Top-Down and Bottom-Up design, Decision tree, decision table, Cohesion, Coupling, Design approaches: Functional and Object- Oriented approach. Function- Oriented Software Design: SA/SD methodology, Structure analysis, DFD, Structure chart, Design review Object-Oriented Software Design: Concept of OO Software – Design and Analysis, Overview of various UML diagrams and UML analysis modeling, analysis case studies, analysis tools, analysis patterns. [8L]

Unit 3: Coding:

Coding standards, code walk- throughs, code inspection, clean room testing and documentation. [4L]

Unit 4: Software Quality Assurance:

Quality Concepts, The Quality Movement, Software Quality Assurance, Software Reviews, Formal Technical Reviews, Formal Approaches to SQA, Statistical Software Quality Assurance, Software Reliability, Mistake Proofing for Software, Introduction to ISO standard.

[7L]

Unit 5: Software Testing Technique:

Software testing fundamentals, Test case design, White-box Testing, Basis path testing, Control structure testing, Black-box testing, Testing for specialized environments, architectures and applications. [6L]

Unit 6: Software Maintenance:

Characteristics, reverse engineering, maintenance process models, estimation of cost. [4L]

MJ-14P: Software Engineering (Lab) *Credits 01 (30 hrs.)*

Program Details:

1. Analysis report of project-
    a. Abstract of Project
    b. Model Description (Best Suitable for Project)
    c. Technology Used
    d. References used for project (Similarities, Differences, Webpages)
2. To perform the system analysis: Requirement analysis, SRS. (According Format)
3. To perform the function oriented diagram: DFD or Structured chart.
4. To perform UML diagrams using UML tool
    a. Introduction of UML
    b. Introduction UML Tool.
5. To perform the user's view analysis: Use case diagram. (Based on given project)
6. To draw the structural view diagram: Class diagram, object diagram. (Based on given project)
7. To draw the behavioral view diagram: Sequence diagram, Collaboration diagram. (Based on given Project)
8. To draw the behavioral view diagram: State-chart diagram, Activity diagram. (Based on given Project)
9. To draw the implementation view diagram: Computer diagram. (Based on given project)

Sample Projects:

1. Passport automation System
2. Book Bank

3. Online Exam Registration
4. Stock Maintenance System
5. Online course reservation system
6. E-ticketing
7. Software Personnel Management System
8. Credit Card Processing
9. E-book management System.
10. Recruitment system

References Books:

1. Pressman, R. S., & Maxim, B. R. (2020). *Software Engineering: A Practitioner' Approach* (9th ed.). McGraw-Hill.
2. Sommerville, I. (2016). *Software Engineering* (10th ed.). Addison-Wesley.
3. Bruegge, B., & Dutoit, A. H. (2010). *Object-Oriented Software Engineering: Using UML, Patterns, and Java* (3rd ed.). Pearson Prentice Hall.
4. Jalote, P. (2006). *An Integrated Approach to Software Engineering* (3rd ed.). Springer.
5. Mall, R. (2018). *Fundamentals of Software Engineering* (5th ed.). PHI Learning.
6. Aggarwal, K. K., & Singh, Y. (2016). *Software Engineering* (3rd rev. ed.). New Age International Publishers.
7. Braude, E. J. (2001). *Software Engineering: An Object-Oriented Perspective*. Wiley.
8. Bell, D. (2005). *Software Engineering for Students* (4th ed.). Addison-Wesley.
9. Schmuller, J. (2004). *SAMS Teach Yourself UML in 24 Hours* (3rd ed.). Sams Publishing.

MJ-15: Compiler Design **Credits 04 (Full Marks: 75)**

OBJECTIVE OF THE COURSE

- Provide an in-depth understanding of the principles and techniques used in compiler design and implementation.
- Cover various phases of a compiler, including lexical analysis, syntax analysis, semantic analysis, optimization, and code generation.
- Teach construction of finite automata and context-free grammars for translating high-level programming languages into machine code.
- Emphasize the role of syntax trees and symbol tables in managing program structure and scope.
- Offer hands-on experience through practical assignments and projects in developing compiler components.
- Utilize compiler construction tools like Lex and Yacc for practical compiler development.
- Explore optimization techniques to improve the efficiency of generated code.
- Delve into error detection and recovery strategies for robust compilation processes.
- Equip students with the skills to design and implement compilers.
- Prepare students for careers in software development and systems programming.

MJ-15T: Compiler Design (Theory) *Credits 03 (45 hrs.)*

Course Content:

Module- I Introduction to compiler **6 Hrs.**

Introduction to Compilers, structure of a compiler, phases of compilation, such as Lexical analysis, code generation, code optimization, table management.

Module- II Syntax Directed Translator **6 Hrs.**

Implementation aspects of programming language constructs such as data elements, Declarations, binding attributes to names, in built data structures such as arrays, records, sets, strings, expression evaluation, statements of different types, parameter passing (call by reference, value, name), storage management, recursion.

Module- III Lexical Analysis **8 Hrs.**

Design of lexical analyzer. Basic parsing techniques such as shift reduce parsing, operator precedence parsing, Top-down parsing, bottom-up parsing. Symbol tables: Contents, data structures, representation of scope. Syntax directed translation.

Module- IV Syntax Analysis **30 Hrs.**

Introduction, The Role of the Parser, Representative Grammars, Context-Free Grammars, Derivations ,Parse Trees and Derivations, Ambiguity , Eliminating Ambiguity , Elimination of Left Recursion , Left Factoring, Top-Down Parsing, Recursive-Descent Parsing, FIRST and FOLLOW, LL(1) Grammars, Nonrecursive Predictive Parsing, Bottom-Up Parsing, Handle Pruning, Shift-

Reduce Parsing, Introduction to LR Parsing: Simple LR, Constructing SLR-Parsing Tables, Viable Prefixes, Canonical LR(1) Items, Constructing LR(1) Sets of Items, Canonical LR(1) Parsing Tables, Constructing LALR Parsing Tables.

Module- V Intermediate-Code Generation **10 Hrs.**

Three-Address Code, Addresses and Instructions, Quadruples, Triples, Static Single-Assignment Form, Code generation, Flow Graph, The Dag representation of Basic Blocks, type checking, cross compiler.

Recommended readings:

1. Santanu Chattopadhyaya, Systems Programming, PHI, 2011.
2. Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, Compilers: Principles, Techniques, and Tools, 2nd edition, Prentice Hall, 2006.
3. D. M. Dhamdhere, Systems Programming, Tata McGraw Hill, 2011.
4. Leland Beck, D. Manjula, System Software: An Introduction to System Programming, 3rd edition, Pearson Education, 2008.
5. Grune D, Van Reeuwijk . K, Bal H. E, Jacobs C J H, Langendoen K, Modern Compiler Design, 2nd edition, Springer, 2012

MJ-15T: Compiler Design (Lab) *Credits 01 (30 hrs.)*

1. Write a Program in LEX/YACC to check whether a given string is a valid ID (Identifier), Keyword, RELOP (Relational Operator) or others.
2. Write a program in LEX/YACC to check whether a given expression (relational or assignment or bitwise operator) is valid or not and it gives the type of expression as output.
3. Write a program in LEX /YACC which takes standard input as output of system date and time and give either of the following messages “Good Morning”, “Good Afternoon”, “Good Evening”
4. Construct a syntax directed translation scheme that translates integers into roman numerals. Implement translator from integers to roman numerals based on above syntax directed translation using LEX/YACC.
5. Write a program using FLEX/YACC, which recognize regular expression.
6. Write a C code analyzer in LEX/YACC: comments, code, white space, count braces, keywords etc. Try to identify function definition and declaration, which are names followed by ‘(‘outside of any braces.
7. Write programs in LEX/YACC, which replaces all the occurrences of “rama” with “RAMA” and “sita” with “SITA”.
8. Write a program in LEX/YACC to check whether a sentence of English language is grammatically correct or not.
9. Write a program in LEX/YACC which takes a English sentence as input and gives the output as the parts of speech.

10. Write a program in LEX/YACC which takes a C program as inputs and delete the comment, white space and Count the no of lines.
11. Write a program in LEX/YACC which counts the no of lines, total no of characters, total no of vowels and total no of punctuation marks in a paragraph.
12. Write a program in LEX/YACC to check the parts of Speech of a Sentence.
13. Write a program in LEX/YACC to count all occurrences of "rama" and "sita" in a given file and eliminate them.
14. Write programs in LEX/YACC that eliminate multiple spaces and tabs and replace with a single space and remove empty lines.
15. Write a lex program to count the number of comment lines in a given C program. Also eliminate them and copy that program into separate file
16. Program to recognize whether a given sentence is simple or compound.
17. Program to recognize a valid arithmetic expression and to recognize the identifiers and operators present. Print them separately.

COMPULSORY RESEARCH METHODOLOGY (CRM)

[for both Hons. and Hons. with Research students]

CRM: Research Methodology and Ethics

CRM (T): Research Methodology and Ethics (Theory) ***Credits 03***

Course Outcome:

Students able to:

CO1: Learn to formulate research problems and design appropriate methodologies.

CO2: Learn to select and apply suitable data collection and sampling techniques.

CO3: Analyze data using descriptive and inferential statistics, create research instruments, and document them in LaTeX.

CO4: Interpret results ethically and present findings in a structured report in LaTex..

Course Content:

Unit 1: Introduction to Research **[6 L]**

Meaning, objectives, and motivation for research, Types of research (Basic vs. Applied, Qualitative vs. Quantitative), Research approaches and scientific method, Criteria of good research

Unit 2: Research Design & Sampling **[8 L]**

Research problem formulation and hypothesis development, Research design: Need, features, and types (Experimental, Descriptive), Sampling techniques: Probability vs. Non-probability, sample size determination

Unit 3: Data Collection & Measurement **[8 L]**

Primary vs. secondary data, Tools: Questionnaires, interviews, observations, Measurement scales (Nominal, Ordinal, Interval, Ratio), Scaling techniques (Likert, Semantic Differential)

Unit 4: Data Analysis & Hypothesis Testing **[8L]**

Descriptive statistics (Mean, SD, Skewness), Inferential statistics: t-test, ANOVA, Chi-square, Correlation and regression analysis, Non-parametric tests (Mann-Whitney, Kruskal-Wallis)

Unit 5: Research Ethics & Report Writing **[5 L]**

Ethical issues in research (Plagiarism, Confidentiality), Structure of a research report, Oral presentation techniques, Use of software (SPSS/R) for data analysis

CRM (P): Research Methodology and Ethics (Lab) ***Credits 01***

List of Programs:

1. LaTeX Installation and First Document
2. Formatting a Research Problem Statement
3. Creating a Research Proposal Template
4. Designing a Sampling Plan with Lists
5. Building a Questionnaire with Tables
6. Creating a Data Codebook
7. Data Import and Cleaning with R/Python
8. Generating Descriptive Statistics
9. Importing Data into LaTeX with csvsimple
10. Formatting Statistical Tables with booktabs
11. Performing and Reporting a T-Test
12. Creating a Correlation Matrix Table
13. Simple Linear Regression Analysis
14. Non-Parametric Mann-Whitney U Test
15. Managing References with BibTeX
16. Creating a Bar Chart with pgfplots
17. Creating a Scatter Plot with pgfplots
18. Plagiarism Check and Ethical Writing
19. Compiling a Full Research Report
20. Designing a Research Poster with beamerposter

Reference Books

1. Kothari, C. R. (2004). Research Methodology: Methods and Techniques (2nd ed.). New Age International Publishers.
2. Creswell, J. W., & Creswell, J. D. (2023). Research Design: Qualitative, Quantitative, and Mixed Methods Approaches (6th ed.). SAGE Publications.
3. Lamport, L. (1994). LaTeX: A Document Preparation System (2nd ed.). Addison-Wesley.
4. Mittelbach, F., & Goossens, M. (2004). The LaTeX Companion (2nd ed.). Addison-Wesley.

MAJOR ELECTIVE (DSE)

Major Elective – 03:

(Image Processing/ Big Data Analytics / Cyber-Security/Mobile Computing)

MJDSE03: Image Processing **Credits 04 (Full Marks: 75)**

Course objectives:

1. To be familiar with Image model, sensing and acquisition, digital image representation, properties of human visual system, various applications.
2. Different image processing operations for improving image quality through enhancement, restoration and filtering etc.
3. Affine transformation and registration compressing data to save storage and channel capacity during transmission. Image segmentation for partitioning into objects and background.

MJDSE-03T: Image Processing (Theory) *Credits 03 (45 hrs.)*

Course Content:

UNIT-I: Introduction: Image definition and its representation, Pixels, Co-ordinate conventions, Image formats (Study of the image matrix), neighbourhood metrics, Sampling and quantization, Types of distance measure (concept only). **(8 hours)**

UNIT-II: Spatial Domain: Image enhancement techniques in spatial domain, Contrast stretching, Histogram Processing, Noise smoothing, Sharpening, Pixel Classification. **(8 hours)**

UNIT-III: Thresholding: Grey level thresholding, global/ local thresholding, Iterative thresholding, Edge detection operators, Region growing, Split/ merge techniques, Image feature/ primitive extraction, Background correction, Color enhancement. **(10 hours)**

UNIT-IV: Image restoration: Basic Framework, Interactive Restoration, Image deformation and geometric transformations, image morphing, Restoration techniques, Noise characterization, Noise restoration filters, Restoration from projections, Hough transform, Huffman coding, Segmentation. **(12 hours)**

UNIT-V: Image Segmentation: Boundary detection based techniques, Point, line detection, Edge detection, Local processing, Regional processing, Region-based segmentation. **(8 hours)**

Reference Books

1. R C Gonzalez, R E Woods, Digital Image Processing, 3rd Edition, Pearson Education.2008.
2. A K Jain, Fundamentals of Digital image Processing, Prentice Hall of India.1989.
3. K R Castleman, Digital Image Processing, Pearson Education.1996

MJDSE-03P: Image Processing (Lab) *Credits 01 (30 hrs.)*

All programs are implemented in Python or any other suitable software.

1. Write a program (in Python) to read and display an image.
2. Write a program (in Python) to convert a color image to gray image.
3. Write a program (in Python) to read, show, and print the properties of an image.
4. Write a program (in Python) to enlarge an image.
5. Write a program (in Python) to rotate an image in clock wise and anticlockwise direction.
6. Write a program (in Python) to implement Color conversion for RGB model to gray scale model.
7. Write a program (in Python) to perform Histogram Equalization.
8. Write a program (in Python) to calculate Histograms.
9. Write a program (in Python) to implement Sharpening (Laplacian Spatial Filters).
10. Write a program (in Python) for Smoothing and Sharpening on a color image.
11. Write a program (in Python) to implement Basic gray-level transforms.
12. Write a program (in Python) to implement Arithmetic and Logical operations on an image.

OR

MJDSE03: Big Data Analytics **Credits 04 (Full Marks: 75)**

Course Outcome (CO):

Students able to:

CO1: Analyze the fundamental concepts, characteristics, and challenges of Big Data in various application domains.

CO2: Design and implement scalable storage solutions for Big Data using distributed file systems and NoSQL databases.

CO3: Develop and optimize data processing workflows using batch and stream processing frameworks such as Hadoop and Spark.

CO4: Apply advanced analytics techniques, including machine learning and data mining, to extract actionable insights from large datasets.

MJDSE-03T: Big Data Analytics (Theory) *Credits 03 (45 hrs.)*

Course Content:

Unit 1: Introduction to Big Data **[6L]**

Definition and characteristics of Big Data (volume, velocity, variety, veracity, value), Big Data ecosystem: hardware, software, and network considerations, Challenges in Big Data: storage, processing, analytics, security, and privacy, Opportunities and applications in domains such as healthcare, finance, social media, and IoT

Unit 2: Big Data Storage **[7L]**

Distributed file systems: HDFS, Google File System, NoSQL databases: key-value stores (e.g., Redis), document databases (e.g., MongoDB), column-family stores (e.g., Cassandra), graph databases (e.g., Neo4j), Data lakes and cloud storage solutions (e.g., AWS S3, Google Cloud Storage), Data formats and serialization techniques: Avro, Parquet, ORC

Unit 3: Big Data Processing **[8L]**

Batch processing frameworks: Hadoop MapReduce, Apache Hive, Apache Pig, In-memory processing with Apache Spark: RDDs, DataFrames, Spark SQL, Stream processing technologies: Apache Kafka, Apache Storm, Apache Flink, Workflow management tools: Apache Oozie, Apache Airflow

Unit 4: Big Data Analytics Techniques **[8L]**

Machine learning for Big Data: supervised and unsupervised learning algorithms, Spark MLlib, Data mining techniques for large datasets: association rule mining, clustering, classification, Graph analytics: algorithms for large graphs, Apache GraphX
Visualization tools for Big Data: Tableau, Power BI, Apache Zeppelin

Unit 5: Applications **[6L]**

Big Data applications in healthcare: patient data analysis, disease prediction, Finance: fraud detection, risk management, social media: sentiment analysis, trend detection, Big Data in IoT: sensor data processing, predictive maintenance, Ethical considerations, data governance, and privacy issues in Big Data

MJDSE-03P: Big Data Analytics (Lab) *Credits 01 (30 hrs.)*

Program Details:

Exercise 1: Installation and configuration of Hadoop ecosystem (HDFS, YARN) on a single-node or pseudo-distributed cluster.

Exercise 2: Perform basic HDFS operations: uploading, downloading, listing, and deleting files.

Exercise 3: Implement a simple Hadoop MapReduce program (e.g., Word Count).

Exercise 4: Use Apache Hive to create tables, load data, and run SQL-like queries on large datasets.

Exercise 5: Implement data processing using Apache Pig scripts.

Exercise 6: Introduction to Apache Spark:

- Load data into RDDs and DataFrames.
- Perform basic transformations and actions.

Exercise 7: Use Spark SQL for structured data analysis.

Exercise 8: Implement a real-time data streaming pipeline using Apache Kafka and Spark Streaming.

Exercise 9: Apply Spark MLlib for a supervised learning task (e.g., classification) on a large dataset.

Exercise 10: Perform association rule mining or clustering on Big Data using Spark MLlib.

Exercise 11: Work with NoSQL databases (MongoDB / Cassandra) for storing and querying data.

Exercise 12: Use a Big Data visualization tool (Tableau / Apache Zeppelin) to create dashboards for insights.

Reference Books:

1. Thomas Erl, Wajid Khattak, Paul Buhler - Big Data Fundamentals: Concepts, Drivers & Techniques, Prentice Hall.
2. Tom White - Hadoop: The Definitive Guide, O'Reilly Media.
3. Bill Chambers, Matei Zaharia - Spark: The Definitive Guide, O'Reilly Media.
4. Foster Provost, Tom Fawcett - Data Science for Business, O'Reilly Media.
5. Jure Leskovec, Anand Rajaraman, Jeffrey Ullman - Mining of Massive Datasets, Cambridge University Press.

OR

MJDSE03: Cyber Security **Credits 04 (Full Marks: 75)**

Course Outcome (CO):

Students able to:

CO1: Analyse cyber threats, vulnerabilities, and attack vectors to assess risks in information systems and networks.
CO2: Design and implement cryptographic protocols and algorithms to ensure data confidentiality, integrity, and authentication.
CO3: Evaluate and apply security mechanisms for protecting network infrastructure, web applications, and cloud-based systems.
CO4: Develop incident response strategies and utilize forensic tools to investigate and mitigate cyber-attacks.

MJDSE-03T: Cyber Security (Theory) *Credits 03 (45 hrs.)*

Course Content:

unit 1: Introduction to cyber security **[6l]**

fundamentals of cyber security: confidentiality, integrity, availability (cia triad), types of cyber threats: malware, phishing, ransomware, social engineering, security policies, risk assessment, and threat modelling, overview of cyber security frameworks: nist, iso/iec 27001, legal and ethical issues in cyber security

unit 2: Cryptography and secure communication **[8l]**

symmetric and asymmetric cryptography: aes, rsa, ecc, hash functions and digital signatures: sha, hmac, public key infrastructure (pki) and certificate authorities, secure communication protocols: tls/ssl, ipsec, ssh, quantum cryptography basics

unit 3: Network security **[7l]**

network vulnerabilities: ddos, man-in-the-middle, packet sniffing, firewalls, intrusion detection and prevention systems (idps), virtual private networks (vpns) and secure tunnelling, network access control: authentication protocols (kerberos, oauth), tools for network security: wireshark, snort, nmap

unit 4: Application and cloud security **[6l]**

web application vulnerabilities: sql injection, xss, csrf, secure software development lifecycle (sdlc), cloud security models: shared responsibility, zero trust architecture, securing cloud environments: aws, azure, google cloud, container security: docker, kubernetes

unit 5: Cyber security operations and forensics **[8l]**

incident response lifecycle: preparation, detection, containment, recovery, digital forensics: evidence collection, chain of custody, forensic tools (autopsy, ftk), malware

analysis: static and dynamic analysis techniques, security information and event management (siem) systems, emerging trends: ai-driven security, blockchain for secure transactions.

MJDSE-03P: Cyber Security (Theory) *Credits 01 (30 hrs.)*

Program Details:

Unit 1: Introduction & Environment Setup

Setting up a virtual lab environment using VirtualBox/VMware, Installing Kali Linux, Wireshark, Nmap, and other security tools, Understanding legal & ethical aspects of penetration testing

Unit 2: Network Scanning & Enumeration

Using Nmap for host discovery & port scanning, Identifying open services and potential vulnerabilities, Banner grabbing & service fingerprinting

Unit 3: Cryptography Implementation

Implementing symmetric encryption (AES) and asymmetric encryption (RSA) in Python, Generating & verifying digital signatures, Hashing techniques: SHA-256, HMAC

Unit 4: Secure Communication Protocols

Setting up TLS/SSL for a web server, Testing SSH-based secure remote access, Exploring IPsec VPN configuration

Unit 5: Web Application Security

Testing for SQL Injection & Cross-Site Scripting (XSS) using DVWA (Damn Vulnerable Web Application), Implementing input validation & secure coding practices, Using Burp Suite for web traffic analysis

Unit 6: Network Security Tools

Configuring and using Snort as an Intrusion Detection System, Setting up firewall rules using ip tables or pfSense, Capturing and analyzing network traffic with Wireshark

Unit 7: Cloud & Container Security

Basics of securing AWS/Azure cloud resources, Configuring IAM roles & policies, Securing Docker containers against vulnerabilities

Unit 8: Digital Forensics & Malware Analysis

Using Autopsy for file system forensics, Basic malware analysis (static & dynamic) in a sandbox, Tracing attack footprints using logs and SIEM tools

References Books:

1. William Stallings – *Cryptography and Network Security: Principles and Practice*, Pearson Education.
2. Michael E. Whitman, Herbert J. Mattord – *Principles of Information Security*, Cengage Learning.
4. Chuck Easttom – *Computer Security Fundamentals*, Pearson Education.
5. Matt Bishop – *Computer Security: Art and Science*, Addison-Wesley.
6. Dafydd Stuttard, Marcus Pinto – *The Web Application Hacker's Handbook*, Wiley.

MINOR (MI)

Minor (MI)-07: Digital Marketing **Credits 04 (Full Marks:75)**

MI-07T: Digital Marketing (Theory) *Credits 04 (60 hrs)*

Course Contents-

Unit 1: Marketing in the Digital World **(12 hours)**

Digital marketing: Concept, Features, Difference between traditional and digital marketing, Moving from traditional to digital Marketing;
Digital Marketing Channels: Intent Based- SEO, Search Advertising; Brand Based-Display Advertising; Community Based-Social Media Marketing; Others-Affiliate, Email, Content, Mobile.

Customer Value Journey: 5As Framework; The Ozone O3 Concept Key; Traits of online consumer

Unit 2: Content and Email Marketing **(8 hours)**

Content Marketing: Step-by-step Content Marketing Developing a content marketing strategy Email Marketing: Types of Emails in email marketing, Email Marketing best practices

Unit 3: Social Media Marketing and Display Marketing **(18 hours)**

Social Media Marketing: Building Successful Social Media strategy; Social Media Marketing Channels; Facebook, LinkedIn, YouTube (Concepts and strategies)

Display Advertising: Working of Display Advertising; Benefits and challenges; Overview of Display ad Process.; Define- Customer, Publisher, Objectives; Format-Budget, Media, Ad Formats, Ad Copy.

Unit 4 Search Engine Marketing **(22 hours)**

Introduction of SEM: Working of Search Engine; SERP Positioning; online search behaviour, DMI's 5P Customer Search Insights Model.
Search Engine Optimization: Overview of SEO Process; Goal Setting-Types.

On-Page Optimization: Keyword Research, SEO Process -Site Structure, Content, Technical Mechanics, Headings, Image & Alt text, Social Sharing, Sitemaps, Technical Aspects-Compatibility, Structured Data Markup.
Off Page Optimisation: Link Formats, Link Building, Content Marketing, Social Sharing; Black and White Hat Techniques
Search Advertising: Overview of PPC Process; Benefits of Paid Search; Basis of Ranking; Goal Setting-Objectives; Account Setting-Creation of Google Ads, Campaign architecture, Campaign setup, Targeting, Bid Strategy, Delivery, Ad Scheduling, Ad Rotation, Keyword Selection; Ad Copy composition, Ad Extension

Suggested Readings:

1. Dodson, I. (2016). *The art of digital marketing: the definitive guide to creating strategic, targeted, and measurable online campaigns*. John Wiley & Sons.
2. Kartajaya, H., Kotler, P., & Setiawan, I. (2016). *Marketing 4.0: moving from traditional to digital*. John Wiley & Sons.
3. Ryan, Damien: *Understanding Digital Marketing - Marketing Strategies for Engaging the Digital Generation*. Kogan Page Limited
4. Moutusy Maity: *Internet Marketing: A practical approach in the Indian Context*: Oxford Publishing
5. Seema Gupta: *Digital Marketing*: Mcgraw Hill
6. *Ultimate guide to digital Marketing* by Digital Marketer

SEMESTER-VIII

MAJOR (MJ)

MJ-16: Artificial Intelligence **Credits 04 (FM:75)**

OBJECTIVE OF THE COURSE

- Provide a comprehensive understanding of the fundamental concepts and techniques in artificial intelligence.
- Introduce key AI topics such as machine learning, natural language processing, computer vision, and robotics.
- Teach problem-solving techniques using search algorithms, logic, and optimization methods.
- Explore knowledge representation and reasoning, including the use of ontologies and semantic networks.
- Emphasize the design and implementation of intelligent agents that can perceive and act in complex environments.
- Cover machine learning models and algorithms, including supervised, unsupervised, and reinforcement learning.
- Develop skills in programming AI applications using languages like Python and AI frameworks such as TensorFlow or PyTorch.
- Discuss ethical and societal implications of AI technologies, including privacy, bias, and accountability.
- Offer hands-on experience through projects and assignments that apply AI techniques to real-world problems.
- Prepare students for advanced study and careers in AI research, development, and application across various industries.

MJ-16T: Artificial Intelligence (Theory) *Credits 03 (45 hrs.)*

Course Contents:

Module- I: Introduction: **6 Hrs.**

Overview and Historical Perspective, Turing test, Physical Symbol Systems and the scope of Symbolic AI, Agents.

.

Module- II: State Space Search: **6 Hrs.**

Depth First Search, Breadth First Search, DFID.

Module- III: Heuristic Search: **8 Hrs.**
Best First Search, Hill Climbing, Beam Search, Tabu Search.

Module- IV: Randomized Search: **4 Hrs.**
Simulated Annealing, Genetic Algorithms, Ant Colony Optimization.

Module- V: Finding Optimal Paths: **8 Hrs.**
Branch and Bound, A*, IDA*, Divide and Conquer approaches, Beam Stack Search.

Module- VI: Problem Decomposition: **8 Hrs.**
Goal Trees, AO*, Rule Based Systems, Rete Net.

Module- VII: Game Playing: **4 Hrs.**
Minimax Algorithm, AlphaBeta Algorithm, SSS*.

Module- VIII: Planning and Constraint Satisfaction: **8 Hrs.**
Domains, Forward and Backward Search, Goal Stack Planning, Plan Space Planning, Graphplan, Constraint Propagation.

Module- XI: Logic and Inferences: **8 Hrs.**
Propositional Logic, First Order Logic, Soundness and Completeness, Forward and Backward chaining.

MJ-16P: Artificial Intelligence (Lab) *Credits 01 (30 hrs.)*

Experiments should include but not limited to:

1. Study of PROLOG.
2. Write the following programs using PROLOG:
3. Write a program to solve 8-queens problem.
4. Solve any problem using depth first search.
5. Solve any problem using best first search.
6. Solve 8- puzzle problem using best first search.
7. Solve Robot (traversal) problem using means End Analysis.
8. Solve Traveling Salesman problem.

Suggested Readings:

1. Deepak Khemani. A First Course in Artificial Intelligence, McGraw Hill Education (India), 2013.
2. Stefan Edelkamp and Stefan Schroedl. Heuristic Search: Theory and Applications, Morgan Kaufmann, 2011.
3. John Haugeland, Artificial Intelligence: The Very Idea, A Bradford Book, The MIT Press, 1985.
4. Pamela McCorduck, Machines Who Think: A Personal Inquiry into the History and Prospects of Artificial Intelligence, A K Peters/CRC Press; 2 edition, 2004.
5. Zbigniew Michalewicz and David B. Fogel. How to Solve It: Modern Heuristics. Springer; 2nd edition, 2004.
6. Judea Pearl. Heuristics: Intelligent Search Strategies for Computer Problem Solving, Addison-Wesley, 1984.
7. Elaine Rich and Kevin Knight. Artificial Intelligence, Tata McGraw Hill, 1991.
8. Stuart Russell and Peter Norvig. Artificial Intelligence: A Modern Approach, 3rd Edition, Prentice Hall, 2009.
9. Eugene Charniak, Drew McDermott. Introduction to Artificial Intelligence, Addison-Wesley, 1985.
10. Patrick Henry Winston. Artificial Intelligence, Addison-Wesley, 1992.

Major Elective – 04:

(Machine Learning/ Natural Language Processing /Data Science)

MJDSE-04: Machine Learning **Credits 04 (Full Marks:75)**

Course Outcome:

Students able to:

CO1: Understand and explain core concepts, types, and real-world applications of machine learning, along with the mathematical foundations of common algorithms.

CO2: Implement and evaluate machine learning models using Python libraries such as Scikit-learn, and gain exposure to advanced tools like TensorFlow or PyTorch.

CO3: Apply machine learning techniques to solve problems across various domains, including data preprocessing, model selection, and interpretation of results.

CO4: Analyze and select appropriate algorithms based on problem type, data characteristics, and performance metrics, while appreciating the role of data, algorithmic thinking, and system design in building effective, scalable, and responsible ML solutions.

MJDSE-04T: Machine Learning (Theory) *Credits 03 (45 hrs)*

Course Content:

Unit 1: Foundations of Machine Learning **[3L]**

What is Machine Learning? Types of ML: Supervised, Unsupervised, Reinforcement Learning, Key ML Terminologies: features, labels, models, training, testing, ML Project Lifecycle and Pipeline Overview

Unit 2: Mathematics for Machine Learning **[3L]**

Linear Algebra: Vectors, matrices, dot products, Probability & Statistics: Mean, variance, Bayes' Theorem, distributions, Calculus: Derivatives, gradients, basics of optimization, Cost and Loss Functions

Unit 3: Data Preprocessing and Exploration **[4L]**

Data collection and loading, Exploratory Data Analysis (EDA), Handling missing data and outliers, Feature engineering and encoding, Data normalization and standardization, Train-test split and validation strategies

Unit 4: Supervised Learning – Regression **[4L]**

Linear Regression (simple & multivariate), Polynomial Regression, Regularization: Ridge, Lasso, ElasticNet, Evaluation metrics: MAE, MSE, RMSE, R^2

Unit 5: Supervised Learning - Classification **[4L]**

Logistic Regression, K-Nearest Neighbors (KNN), Decision Trees, Support Vector Machines (SVM), Evaluation metrics: Accuracy, Precision, Recall, F1 Score, Confusion Matrix, ROC-AUC

Unit 6: Ensemble Learning Methods **[4L]**

Bagging vs Boosting, Random Forest, Gradient Boosting Machines (GBM), XGBoost, LightGBM, CatBoost, Feature importance and interpretability

Unit 7: Unsupervised Learning **[5L]**

Clustering: K-Means, DBSCAN, Hierarchical Clustering, Dimensionality Reduction: PCA, t-SNE, UMAP, Anomaly Detection and Applications, Association Rule Learning (Apriori, Eclat)

Unit 8: Model Evaluation and Optimization **[3L]**

Cross-validation (K-Fold, Stratified), Hyperparameter tuning: Grid Search, Random Search, Bias-Variance Tradeoff, Underfitting vs Overfitting, Model selection strategies

Unit 9: Intro to Neural Networks (Bridge to Deep Learning) **[3L]**

Perceptron and Multi-layer Perceptron (MLP), Forward and Backpropagation, Activation Functions, Use cases: classification, regression, Brief intro to TensorFlow or PyTorch

Unit 10: Time Series and Sequential Data **[2L]**

Time series components: trend, seasonality, noise, Lag features and rolling statistics, ARIMA and forecasting basics, Evaluating forecasting models

MJDSE-04P: Machine Learning (Lab) *Credits 01 (30 hrs)*

Program Details:

Unit 1: Python & ML Environment Setup-

Setting up Python environment for ML (Anaconda, Jupyter Notebook, VS Code), Introduction to NumPy, Pandas, Matplotlib, Seaborn for data handling & visualization.

Unit 2: Data Preprocessing & EDA-

Importing datasets and performing exploratory data analysis (EDA), Handling missing data, outliers, categorical encoding, normalization, and standardization.

Unit 3: Regression Models

Implementing Simple & Multiple Linear Regression, Implementing Polynomial Regression and regularization techniques (Ridge, Lasso).

Unit 4: Classification Models

Implementing Logistic Regression and K-Nearest Neighbors (KNN), Implementing Decision Trees and Support Vector Machines (SVM) with evaluation metrics.

Unit 5: Ensemble Learning

Implementing Random Forest and Gradient Boosting (XGBoost/LightGBM), Feature importance analysis and visualization.

Unit 6: Unsupervised Learning

Implementing K-Means and DBSCAN clustering with visualization, Performing Principal Component Analysis (PCA) and t-SNE for dimensionality reduction.

Unit 7: Model Optimization

Hyperparameter tuning with Grid Search and Random Search, Cross-validation and performance comparison of models.

Unit 8: Intro to Neural Networks

Implementing a simple Multi-layer Perceptron (MLP) using TensorFlow/Keras.

Unit 9: Time Series Analysis

Building a simple ARIMA forecasting model and evaluating results.

Unit 10: Model Deployment

Saving and loading ML models with Pickle/Joblib, Deploying an ML model using Flask or FastAPI.

Suggested Readings:

1. Christopher M. Bishop – Pattern Recognition and Machine Learning – Springer, 2006.
2. Trevor Hastie, Robert Tibshirani, Jerome Friedman – The Elements of Statistical Learning: Data Mining, Inference, and Prediction – Springer, 2nd Edition, 2009.
3. Kevin P. Murphy – Machine Learning: A Probabilistic Perspective – MIT Press, 2012.
4. Ian Goodfellow, Yoshua Bengio, Aaron Courville – Deep Learning – MIT Press, 2016.
5. Aurélien Géron – Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow O'Reilly Media, 3rd Edition, 2022.

MJDSE-04: Natural Language Processing **Credits 04 (Full Marks:75)**

Students able to:

CO1: Learn to create CORPUS linguistics based on digestive approach (Text Corpus method)

CO2: Learn method for statistical approaches to machine translation.

CO3: Demonstrate the state-of-the-art algorithms and techniques for text-based processing of natural language with respect to morphology.

CO4: Develop Statistical Methods for Real World Applications and explore deep learning based NLP.

MJDSE-04T: Natural Language Processing (Theory) *Credits 03 (45 hrs)*

Course Content:

Unit 1: Introduction to NLP **[3L]**

Introduction to NLP - Various stages of NLP –The Ambiguity of Language: Why NLP Is Difficult Parts of Speech: Nouns and Pronouns, Words: Determiners and adjectives, verbs, Phrase Structure. Statistics Essential Information Theory: Entropy, perplexity, the relation to language, Cross entropy.

Unit 2: Text Pre-processing and Morphology **[5L]**

Character Encoding, Word Segmentation, Sentence Segmentation, Introduction to Corpora, Corpora Analysis. Inflectional and Derivation Morphology, Morphological analysis and generation using Finite State Automata and Finite State transducer.

Unit 3: Language Modelling **[6L]**

Words: Collocations- Frequency-Mean and Variance – Hypothesis testing: The t test, Hypothesis testing of differences, Pearson‘s chi-square test, Likelihood ratios. Statistical Inference: n -gram Models over Sparse Data: Bins: Forming Equivalence Classes- N gram model - Statistical Estimators- Combining Estimators.

Unit 4: Word Sense Disambiguation **[5L]**

Methodological Preliminaries, Supervised Disambiguation: Bayesian classification, An information theoretic approach, Dictionary-Based Disambiguation: Disambiguation based on sense, Thesaurus based disambiguation, Disambiguation based on translations in a second-language corpus.

Unit 5: Markov Model and POS Tagging **[5L]**

Markov Model: Hidden Markov model, Fundamentals, Probability of properties, Parameter estimation, Variants, Multiple input observation. The Information Sources in Tagging: Markov model taggers, Viterbi algorithm, Applying HMMs to POS tagging, Applications of Tagging.

Unit 6: Probabilistic Context Free Grammars and Probabilistic parsing [5L]

The Probability of a String, Problems with the Inside-Outside Algorithm, Parsing for disambiguation, Tree banks, Parsing models vs. language models, Phrase structure grammars and dependency, Lexicalized models using derivational histories, Dependency-based models.

Unit 7: Syntax and Semantics [6L]

Shallow Parsing and Chunking, Shallow Parsing with Conditional Random Fields (CRF), Lexical Semantics, Word Net, Thematic Roles, Semantic Role Labelling with CRFs. Statistical Alignment and Machine Translation, Text alignment, Word alignment, Information extraction, Text mining, Information Retrieval, NL interfaces, Sentimental Analysis, Question Answering Systems, Social network analysis.

MJDSE-04P: Natural Language Processing (Lab) *Credits 01 (30 hrs)*

Program Details:

1. **Text Preprocessing** — Implement tokenization, stemming, lemmatization, and stopword removal using Python (NLTK/spaCy).
2. **Corpus Creation and Analysis** — Build a small text corpus and perform frequency analysis, collocation extraction, and concordance generation.
3. **Morphological Analysis** — Implement finite state automata for basic morphological analysis.
4. **Statistical Language Modelling** — Build unigram, bigram, and trigram models and evaluate perplexity.
6. **POS Tagging with Rule-Based Method** — Create a simple POS tagger using a lexicon and rules.
7. **POS Tagging with HMMs** — Implement a Hidden Markov Model POS tagger and apply the Viterbi algorithm.
8. **Word Sense Disambiguation (Supervised)** — Train a classifier (e.g., Naive Bayes) for word sense disambiguation.
9. **Word Sense Disambiguation (Dictionary-Based)** — Implement Lesk's algorithm for WSD.
10. **Probabilistic Parsing** — Implement a basic probabilistic context-free grammar (PCFG) parser.
11. **Named Entity Recognition (NER)** — Perform NER using spaCy or CRF-based models.
12. **Sentiment Analysis** — Implement sentiment classification using logistic regression or an LSTM-based deep learning model.
13. **Machine Translation (Basic)** — Implement a statistical alignment model for word/phrase alignment (IBM Model 1 or basic translation memory).
14. **Question Answering System** — Develop a simple fact-based QA system using text retrieval and extraction techniques.

Suggested Readings:

1. Christopher D. Manning and Hinrich Schutze, — Foundations of Natural Language Processing‖ , 6th Edition, The MIT Press Cambridge, Massachusetts London, England, 2003
2. Daniel Jurafsky and James H. Martin —Speech and Language Processing‖, 3rd edition, Prentice Hall, 2009.
3. Nitin Indurkhya, Fred J. Damerau —Handbook of Natural Language Processing‖, Second Edition, CRC Press, 2010.
4. James Allen —Natural Language Understanding‖, Pearson Publication 8th Edition. 2012.
5. Chris Manning and Hinrich Schütze, —Foundations of Statistical Natural Language Processing‖, 2nd edition, MIT Press Cambridge, MA, 2003.
6. Hobson lane, Cole Howard, Hannes Hapke, —Natural language processing in action‖ MANNING Publications, 2019.
7. Alexander Clark, Chris Fox, Shalom Lappin, —The Handbook of Computational Linguistics and Natural Language Processing‖, Wiley-Blackwell, 2012
8. Rajesh Arumugam, Rajalingappa Shanmugamani —Hands-on natural language processing with python: A practical guide to applying deep learning architectures to your NLP application‖. PACKT publisher, 2018.
9. Christopher D. Manning and Hinrich Schütze – *Foundations of Statistical Natural Language Processing*, MIT Press, 2003.
10. Daniel Jurafsky and James H. Martin – *Speech and Language Processing*, 3rd Edition, Prentice Hall, 2009.
11. Nitin Indurkhya, Fred J. Damerau – *Handbook of Natural Language Processing*, 2nd Edition, CRC Press, 2010.
12. Hobson Lane, Cole Howard, Hannes Hapke – *Natural Language Processing in Action*, Manning Publications, 2019.
13. Rajesh Arumugam, Rajalingappa Shanmugamani – *Hands-On Natural Language Processing with Python*, Packt Publishing, 2018.

MJDSE-04: Data Science **Credits 04 (Full Marks:75)**

Course Outcome:

Students able to:

CO1: Learn, analyze and interpret data using appropriate tools and techniques to derive actionable insights from structured and unstructured datasets.

CO2: Apply programming skills, especially in Python and libraries like Pandas, NumPy, and Scikit-learn, to solve real-world data-driven problems.

CO3: Build and evaluate predictive models using machine learning algorithms, while understanding the mathematical foundations behind them.

CO4: Demonstrate awareness of ethical, legal, and societal implications of data use, including privacy.

MJDSE-04T: Data Science (Theory) *Credits 03 (45 hrs)*

Course Content:

Unit 1: Introduction to Data Science **[3L]**

What is Data Science? Applications and Use Cases, Data Science Workflow, Roles: Data Scientist vs Analyst vs Engineer

Unit 2: Data Wrangling and Exploration **[3L]**

Data Types and Structures (CSV, JSON, SQL, etc.), Pandas and NumPy for data manipulation, Handling Missing Data and Outliers, Exploratory Data Analysis (EDA), Data Visualization (Matplotlib, Seaborn)

Unit 3: Statistics and Probability **[3L]**

Descriptive Statistics (mean, median, std. dev), Probability Distributions (normal, binomial, Poisson), Bayes' Theorem, Inferential Statistics: Hypothesis Testing, Confidence Intervals, Correlation and Causation

Unit 4: Programming for Data Science **[4L]**

Python Basics for Data Science, Functions, Loops, List Comprehensions, Writing Clean, Modular Code, Jupyter Notebooks & IDEs, Version Control with Git

Unit 5: Machine Learning Foundations **[5L]**

Supervised vs Unsupervised Learning, Model Evaluation (Train/Test Split, Cross-Validation), Metrics (Accuracy, Precision, Recall, F1 Score), Linear Regression, Logistic Regression, Decision Trees and Random Forests

Unit 6: Advanced Machine Learning **[5L]**

Support Vector Machines (SVM), Clustering (K-Means, DBSCAN), Ensemble Methods, (Boosting, Bagging), Dimensionality Reduction (PCA, t-SNE), Model Tuning and Hyperparameter Optimization (Grid Search, Random Search)

Unit 7: Feature Engineering and Model Deployment **[4L]**

Feature Scaling, Encoding Categorical Variables, Handling Imbalanced Data, Pipelines and Model Persistence (Pickle, Joblib), Model Deployment Basics (Flask, FastAPI), Introduction to MLOps Concepts

Unit 8: Time Series and Forecasting **[4L]**

Time Series Components and Decomposition, ARIMA, SARIMA Models, Seasonality and Trend Analysis, Forecasting Accuracy Metrics (MAE, RMSE), Handling Temporal Data in Pandas

Unit 9: Deep Learning Basics **[4L]**

Neural Networks Overview, Backpropagation and Activation Functions, Introduction to TensorFlow/Keras or PyTorch, CNNs, RNNs (very high-level intro), Use Cases (Image Classification, NLP)

MJDSE-04P: Data Science (Lab) *Credits 01 (30 hrs)*

1. Anaconda Environment Setup and Jupyter Notebook Basics
2. Data Import from CSV and JSON Files
3. Connecting to and Querying a SQL Database
4. Data Cleaning: Handling Missing Values and Duplicates
5. Data Transformation with Pandas
6. Exploratory Data Analysis (EDA) with Descriptive Statistics
7. Data Visualization with Matplotlib: Histograms and Box Plots
8. Advanced Visualization with Seaborn: Heatmaps and Pair Plots
9. Calculating and Interpreting Correlation Matrices
10. Implementing and Sampling from Probability Distributions
11. Performing a Statistical Hypothesis Test (T-Test)
12. Building and Evaluating a Linear Regression Model
14. Implementing and Tuning a Decision Tree Classifier
15. Building a Random Forest Model
16. Unsupervised Learning with K-Means Clustering
17. Feature Engineering: Scaling and Encoding
18. Saving and Loading Models with Joblib
19. Creating a Simple Flask API for Model Deployment

20. Time Series Analysis and Forecasting with ARIMA
21. Building a Basic Neural Network with TensorFlow/Keras

Suggested Readings:

1. Wes McKinney, Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython, O'Reilly Media.
2. Jake VanderPlas, Python Data Science Handbook: Essential Tools for Working with Data, O'Reilly Media.
3. Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly Media.
4. Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer.
5. Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, An Introduction to Statistical Learning with Applications in R and Python, Springer.
6. Wes McKinney, Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython, O'Reilly Media.
7. Jake VanderPlas, Python Data Science Handbook: Essential Tools for Working with Data, O'Reilly Media.
8. Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly Media.
9. Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer.
10. Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, An Introduction to Statistical Learning with Applications in R and Python, Springer.

MAJOR ELECTIVE- 05

[to be studied by Honours students only]

(Advanced DBMS / Green Computing / Cryptography and Steganography)

MJDSE-05: Advanced DBMS **Credits 04 (Full Marks: 75)**

Course Outcome:

Students able to:

CO1: Comprehend and analyze different database models, user types, and system architectures.

CO2: Apply advanced relational database concepts, including dependencies, normalization, and chase algorithms.

CO3 Understand and implement parallel, distributed, active, and deductive database systems, including transaction processing protocols.

CO4: Model complex data using object-oriented and object-relational databases, and analyze real-world case studies for practical applications.

MJDSE-05T: Advanced DBMS (Theory) *Credits 03 (45 hrs)*

Course Content:

Unit 1: Introduction to Advanced Database Concepts **[3L]**

Concept and overview of DBMS, database models, database languages, database users, DBA functions, and the three-schema architecture.

Unit 2: Relational Databases and Integrity Constraints **[5L]**

Functional, multivalued, and join dependencies; template algebraic, inclusion and generalized functional dependencies; chase algorithms; synthesis of relational schemes.

Unit 3: Query Processing and Optimization **[5L]**

Evaluation of relational operations, transformation of expressions, indexing, optimization strategies, null values, and partial information.

Unit 4: Parallel and Distributed Databases **[5L]**

Fragmentation and replication, location and fragment transparency, distributed query processing and optimization, distributed transactions and concurrency control, deadlock handling, commit protocols, and parallel query evaluation.

Unit 5: Advanced Transaction Processing **[5L]**

Nested and multilevel transactions, compensating transactions, sagas, long-duration transactions, weak consistency levels, transaction workflows, processing monitors.

Unit 6: Active Databases **[3L]**

Triggers in SQL, ECA rules (Event-Condition-Action), concurrency control, compensation, recovery.

Unit 7: Deductive Databases **[4L]**

Datalog and recursion, evaluation strategies, recursive queries with negation.

Unit 8: Object-Oriented and Object-Relational Databases **[5L]**

Modeling complex semantics: specialization, generalization, aggregation, object identity and references, architecture of object-oriented and object-relational databases.

Case studies: Gemstone, O2, Object Store, SQL3, Oracle XXI, DB2.

MJDSE-05P: Advanced DBMS (Lab) *Credits 01 (30 hrs)*

Program Details:

Unit 1: Structured Query Language (SQL):

Creating a database, Creating tables and specifying data types, Specifying constraints (primary key, foreign key, check, not null), Creating indexes

Unit 2: Table and Record Handling:

Using INSERT, SELECT, DELETE, UPDATE, TRUNCATE, DROP, ALTER statements

Unit 3: Retrieving Data from Database:

SELECT with WHERE, logical operators, Using IN, BETWEEN, LIKE, ORDER BY, GROUP BY, HAVING, Aggregate functions (SUM, COUNT, AVG, MIN, MAX),
Table joins: INNER JOIN, OUTER JOIN, SELF JOIN, Subqueries and nested queries

Unit 4: Database Management:

Creating and using views, Using column aliases, Creating users, Granting and revoking privileges

Unit 5: Advanced PL/SQL (Oracle):

Using cursors in PL/SQL, Writing stored procedures in PL/SQL

Suggested Readings:

1. Abraham Silberschatz, Henry Korth, and S. Sudarshan – Database System Concepts, McGraw-Hill.
2. Raghu Ramakrishnan – Database Management Systems, WCB/McGraw-Hill.
3. Bipin Desai – An Introduction to Database Systems, Galgotia.
4. J. D. Ullman – Principles of Database Systems, Galgotia.
5. R. Elmasri and S. Navathe – Fundamentals of Database Systems, Addison-Wesley.
6. Serge Abiteboul, Richard Hull, and Victor Vianu – Foundations of Databases, Addison-Wesley.
7. Abraham Silberschatz, Henry Korth, and S. Sudarshan – Database System Concepts, McGraw-Hill.
8. Raghu Ramakrishnan – Database Management Systems, WCB/McGraw-Hill.
9. Bipin Desai – An Introduction to Database Systems, Galgotia.
11. J. D. Ullman – Principles of Database Systems, Galgotia.
12. R. Elmasri and S. Navathe – Fundamentals of Database Systems, Addison-Wesley.
13. Serge Abiteboul, Richard Hull, and Victor Vianu – Foundations of Databases, Addison-Wesley.

OR

MJDSE-05: Green Computing **Credits 04 (Full Marks: 75)**

Course Outcome:

Students able to:

CO1: Discuss Green IT concepts, its dimensions, strategies, and sustainability trends.

CO2: Describe green devices, hardware, and software methodologies for energy-efficient computing.

CO3: Explain green enterprise activities, functions, and management practices along with necessary components of Green IT.

CO4: Discuss laws, standards, and protocols regulating Green IT and their role in sustainable IT practices.

MJDSE-05T: Green Computing (Theory) *Credits 03 (45 hrs)*

Course Content:

Unit 1: Fundamentals: **[10L]**

Green IT Fundamentals: Business, IT, and the Environment – Green computing: carbon foot print, Green IT Strategies: Drivers, Dimensions, and Goals – Environmentally Responsible Business: Policies, Practices, and Metrics. Environmental Concerns and Sustainable Development, Environmental Impacts of IT, Holistic Approach to Greening IT, Applying IT for enhancing environmental sustainability, Green IT Standards and Eco-Labelling of IT.

Unit 2: Green assets and modelling: **[6L]**

Green Assets: Buildings, Data Centres, Networks, and Devices – Green Business Process Management: Modeling, Optimization, and Collaboration – Green Enterprise Architecture – Environmental Intelligence – Green Supply Chains – Green Information Systems: Design and Development Models.

Unit 3: Green Hardware and Software: **[6L]**

Introduction, Life Cycle of a device or hardware, Reuse, Recycle and Dispose. Green Software: Introduction, Energy-saving software techniques, Evaluating and Measuring software Impact to platform power.

Unit 4: Grid Framework: **[8L]**

Virtualization of IT systems – Role of electric utilities, Telecommuting, teleconferencing and teleporting – Materials recycling – Best ways for Green PC – Green Data centre – Green Grid framework.

Unit 5: Green Compliance: **[5L]**

Socio-cultural aspects of Green IT – Green Enterprise Transformation Roadmap – Green Compliance: Protocols, Standards, and Audits – Emergent Carbon Issues: Technologies and Future.

MJDSE-05P: Green Computing (Lab) *Credits 01 (30 hrs)*

Program Details:

1. Working as Motivators under the Swatch Bharat Campaign of the Government.
2. Literacy drive : Teaching in the Charitable School
3. Enroll as NSS Volunteers for various projects (Cleanliness, Women/child health awareness).
4. Counseling camps in villages
5. Tree plantation
6. Enroll in the Gandhian Studies Centre as student Volunteer for surveys in villages.
7. Women Empowerment Programmes.
8. Environment Awareness (Reduce Pollution)
9. Disaster Management/Relief Work.

Suggested Readings:

1. Harnessing Green IT Principles and Practices , San Murugesan, G.R. Gangadharan Wiley Publication, ISBN:9788126539680
2. Bhuvan Unhelkar,—Green IT Strategies and Applications-Using environmental Intelligence, CRC Press, June 2014.
3. Woody Leonhard, Katherine Murray, —Green Home computing for dummies, August 2012.

OR

MJDSE-05: Cryptography and Steganography **Credits 04 (Full Marks: 75)**

Course Outcome:

Students able to:

CO1: Ability to analyze and determine for any organization the security requirements and appropriate solutions., Ability to protect system from different types of threats, malicious software's vulnerabilities and attacks.

CO2: Ability to describe symmetric and public key encryption algorithms like DES, AES, RSA etc.

CO3: Ability to identify ethical, professional responsibilities, risks and liabilities in computer and network environment, and best practices to write security policy. Ability to narrate the Authentication of digital certificates.

CO4: Ability to differentiate MAC and hashing techniques needed for authentication. Ability to protect information from different types of threats, malicious software's vulnerabilities and attacks through Data hiding and Watermarking scheme.

MJDSE-05T: Cryptography and Steganography (Theory) *Credits 03 (45 hrs)*

Course Content:

Unit 1: Introduction

Introduction to security attacks, services and mechanism, introduction to cryptography. Conventional Encryption: Conventional encryption model, classical encryption techniques-substitution ciphers and transposition ciphers, cryptanalysis, stereography. [6L]

Unit 2: Ciphers

Stream and block ciphers. Modern Block Ciphers: Block ciphers principals, Shannon's theory of confusion and diffusion, feistal structure, data encryption standard (DES), strength of DES, differential and linear crypt analysis of DES. AES. [6L]

Unit 3: Digital Signature:

Digital Signature, Authentication, Key Management [5]

Unit 4: Introduction to Information Hiding

Technical Steganography, Linguistic Steganography, Copy Right Enforcement, Wisdom from Cryptography Principles of Steganography: Framework for Secret Communication, Security of Steganography System, Information Hiding in Noisy Data , Adaptive versus non-Adaptive Algorithms, Active and Malicious Attackers, Information hiding in Written Text. [5L]

Unit 5: Steganographic Techniques

Substitution systems and Bit Plane Tools, Transform Domain Techniques: - Spread Spectrum and Information hiding, Statistical Steganography, Distortion Techniques, Cover Generation Techniques.

Unit 6: Steganalysis:
Looking for Signatures: - Extracting hidden Information, Disabling Hidden Information. [10L]

Unit 7: Watermarking and Copyright Protection
Basic Watermarking, Watermarking Applications, Requirements and Algorithmic Design Issues, Evaluation and Benchmarking of Watermarking system. [8L]

MJDSE-05P: Cryptography and Steganography (Lab) *Credits 01 (30 hrs)*

Program Details:

Suggested Readings:

1. William Stallings, —Cryptography and Network Security: Principals and Practice‖, Prentice Hall, New Jersy.
2. Johannes A. Buchmann, —Introduction to Cryptography‖, Springer-Verlag.
4. Bruce Schiener, —Applied Cryptography‖.
5. Katzendbisser, Petitcolas, " Information Hiding Techniques for Steganography and Digital Watermarking", Artech House.
6. Peter Wayner, "Disappearing Cryptography: Information Hiding, Steganography and Watermarking 2/e", Elsevier
7. Bolle, Connell et. al., "Guide to Biometrics", Springer

MAJOR ELECTIVE- 06

[to be studied by Honours students only]

(Soft Computing / Quantum Computing/ Real Time System)

MJDSE-06: Soft Computing **Credits 04 (Full Marks: 75)**

MJDSE-06T: Soft Computing (Theory) *Credits 03 (45 hrs)*

Course Content:

Unit 1: Introduction to Soft Computing **[4 L]**

Soft computing vs. hard computing, evolution of soft computing, features and types of soft computing, applications of soft computing, basics of machine learning.

Unit 2: Neural Networks and Back Propagation networks **[8 L]**

Basic concepts of Neural Networks, Model of Artificial Neuron, Neural Network Architectures, Characteristics of neural networks, Learning Methods, Early neural network architectures, Application domains. Back propagation network (BPN), Back propagation Learning, Applications of BPN, Parameter selection, Variations of Back propagation Algorithms.

Unit 3: Fuzzy Sets and Fuzzy Relations **[5 L]**

Fuzzy versus Crisp, Crisp Sets, Fuzzy sets, Membership functions, fuzzy set operations, properties of Fuzzy sets, Crisp Relations, Fuzzy relations – Fuzzy Cartesian product, Operations of Fuzzy Relations.

Unit 4: Genetic Algorithms **[7 L]**

History of Genetic Algorithm, Basic concepts, Creation of offspring, working principles, encoding, fitness function, reproduction, Genetic modelling: Inheritance operator, crossover, inversion & deletion, mutation operator, Bitwise operator, Generational Cycle, Convergence of GA, Applications & advances in GA, Differences & similarities between GA & other traditional method, Hybrid systems, evolutionary computing, Genetic

Algorithm based on Back propagation networks- Implementation and comparison on performance of traditional algorithms with Genetic Algorithm.

MJDSE-06P: Soft Computing (Practical) *Credits 01 (30 hrs)*

1. **Perceptron Implementation** — Design and train a single-layer perceptron for binary classification.

2. **Multi-Layer Perceptron (MLP)** — Implement an MLP with Back Propagation for function approximation.
3. **Fuzzy Set Operations** — Implement fuzzy union, intersection, complement, and Cartesian product.
4. **Defuzzification Methods** — Implement and compare defuzzification techniques (Centroid, Mean of Maxima, etc.).
5. **Genetic Algorithm Basics** — Implement GA to solve a simple optimization problem (e.g., function maximization).
6. **GA for Combinatorial Problems** — Apply GA to solve the Travelling Salesman Problem (TSP).
7. **Hybrid GA–BPN System** — Train a neural network using GA for weight optimization and compare with standard backpropagation.

Suggested Readings:

1. S, Rajasekaran & G.A. Vijayalakshmi Pai, —Neural Networks, Fuzzy systems and evolutionary algorithms: Synthesis and Applications‖, PHI Publication, 2nd Ed. 2017.
2. Timothy J. Ross, —Fuzzy Logic with Engineering Applications‖, John Wiley and Sons, 3rd ed, 2011.
3. 3. S.N. Sivanandam & S.N. Deepa, —Principles of Soft Computing‖, Wiley Publications, 3rd ed, 2018.
4. Jang, Jyh-Shing Roger, Chuen-Tsai Sun, and Eiji Mizutani. "Neuro-fuzzy and soft computing a computational approach to learning and machine intelligence" Pearson, 1997.

OR

MJDSE-06: Quantum Computing **Credits 04 (Full Marks: 75)**

Course Outcome (CO):

Students able to:

CO1: Explain quantum mechanics foundations (qubits, superposition, entanglement) and contrast classical vs. quantum computational models.

CO2: Design and implement quantum circuits using gates (Hadamard, CNOT, Toffoli) and simulate algorithms (Deutsch-Jozsa, Grover's) on quantum SDKs (Qiskit/Cirq).

CO3: Analyze quantum algorithms (Shor's, QFT, VQE) and evaluate their speedup over classical counterparts for cryptography, optimization, and chemistry.

CO4: Assess hardware challenges (decoherence, error correction) and emerging technologies (superconducting, photonic, trapped-ion qubits).

MJDSE-06T: Quantum Computing (Theory) *Credits 03 (45 hrs)*

Course Content:

Unit 1: Introduction **[6L]**

Quantum Foundations: Qubits, Bloch sphere, Dirac notation, Postulates of quantum mechanics Entanglement and Bell inequalities

Unit 2: Quantum Circuits **[10L]**

Quantum Circuits & Gates: Universal gate sets (Pauli, Hadamard, CNOT), Quantum teleportation and superdense coding

Unit 3: Algorithms **[7L]**

Quantum Algorithms: Deutsch-Jozsa, Bernstein-Vazirani, Grover's search, Shor's factoring

Unit 4: Quantum Techniques **[7L]**

Variational Quantum Eigensolver (VQE), Quantum Machine Learning (QML) basics Lab: VQE for molecular ground states

Unit 5: Advanced Quantum **[5L]**

Decoherence, T1/T2 times, Surface codes, fault-tolerant QC

MJDSE-06P Quantum Computing (Lab) *Credits 01 (30 hrs)*

Program Details:

Unit 1: Quantum Basics in Practice

Qubit representation and initialization in Qiskit/Cirq, Applying single-qubit gates (X, Y, Z, H, S, T) and visualizing on Bloch sphere, Multi-qubit operations: CNOT, SWAP, Toffoli gates

Unit 2: Quantum States & Measurement

Creating superposition states and entangled states (Bell states, GHZ states), Measurement in different bases, Verification of entanglement via Bell inequality tests

Unit 3: Algorithm Implementations

Deutsch-Jozsa algorithm implementation and verification, Grover‘s search algorithm for small database problems, Bernstein–Vazirani algorithm

Unit 4: Variational & Learning Approaches

Implementing Variational Quantum Eigensolver (VQE) for simple molecular Hamiltonians, Intro to Quantum Machine Learning (QML) with basic classification example,

Unit 5: Hardware & Noise Considerations

Running circuits on IBM Quantum Experience or other cloud-based QPUs, Exploring noise models and simple error mitigation techniques, Comparing simulated vs real hardware results

Suggested Readings:

1. "Quantum Computation and Quantum Information" – Michael Nielsen & Isaac Chuang
2. —Quantum Computing for Computer Scientists" – *Noson S. Yanofsky & Mirco A. Mannucci*
3. Michael A. Nielsen & Isaac L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press.
4. Noson S. Yanofsky & Mirco A. Mannucci, *Quantum Computing for Computer Scientists*, Cambridge University Press.
5. Chris Bernhardt, *Quantum Computing for Everyone*, MIT Press.
6. Jack D. Hidary, *Quantum Computing: An Applied Approach*, Springer.
7. Robert S. Sutor, *Dancing with Qubits*, Packt Publishing.

OR

MJDSE-06: Real Time System **Credits 04 (Full Marks: 75)**

Course Outcome:

Students able to:

CO1: Learn to design and manage real-time systems by applying concepts of task scheduling, process management, and synchronization.

CO2: Students will be able to analyze and optimize real-time systems using various scheduling algorithms and fault-tolerant techniques.

CO3: Students will develop expertise in creating real-time applications and managing inter-process communication.

CO4: Students will acquire experience with real-time system modeling and case studies like RTLinux and Windows.

MJDSE-06T: Real Time System (Theory) *Credits 03 (45 hrs)*

Course Content:

Unit 1: Introduction to Real Time System **[3L]**

Historical background, Elements of a Computer Control System, RTS- Definition, Issues in Real-Time Computation, Structure of a Real-time System, Characterizing RTS and Tasks, Time Constraints, Performance measure for RTS.

Unit 2: Real Time Task Assignment and Scheduling **[8L]**

Classical uniprocessor scheduling of IRIS tasks, Common approaches to RT scheduling: clock driven, weighted Round Robin, Priority driven, Earliest Deadline First, Least Slack Time First and Rate Monitoring algorithm. Offline vs Online Scheduling, Scheduling Aperiodic and Sporadic jobs in Priority Driven and Clock driven Systems.

Unit 3: Handling Resource Sharing in RTS **[8L]**

Effect of Resource Contention and Resource Access Control (RAC), Non-preemptive critical sections, Basic priority Inheritance and Priority Ceiling protocols (PCP), Stack based PCP, Use of PCP in Dynamic Priority Systems. Preemption Ceiling Protocol, Access Control in Multiple-Module resources. Controlling concurrent access to data objects.

Unit 4: Scheduling RT tasks in Multiprocessor and Distributed Systems **[4L]**

Multiprocessor and Dynamic task allocation, Fault tolerant Scheduling, Clock Synchronizations.

Unit 5: Real-Time Databases **[4L]**

Real-time vs General Purpose Databases, Main memory Databases, Transaction priorities, Transaction aborts, Concurrency control issues, Disk Scheduling algorithms, Two-phase approach to improve predictability, Maintaining Serialization Consistency, Databases for Hard RTS.

Unit 6: Real-Time Communication **[4L]**

Basic concepts of RT communication, Soft and Hard RT communication systems, Model of RT communication, RT communications in LAN, Boundary Access Protocols for LANs, RT communications over Packet Switched networks, QoS, Rate Control, Internet and Resource Reservation Protocols.

Unit 7: Real-Time OS **[4L]**

Time Services, Features of RTOS, Time services, UNIX and Windows as RTOS, POSIX, Survey of contemporary RTOS viz. PSOS, VRTX, RT Linux, Lynx, Windows CE etc., Benchmarking RTS, Rhealstone Metric, Interrupt processing overhead, Tri-dimensional measure, Determining kernel preemptibility.

MJDSE-06P Quantum Computing (Lab) *Credits 01 (30 hrs)*

Program Details:

Lab 1: Introduction to RTOS & Task Creation

Setup FreeRTOS/RTLinux. Write a periodic and an aperiodic task with deadlines.

Lab 2: Real-Time Scheduling Basics

Implement **Round Robin** and **Priority Scheduling**. Compare results with theoretical expectations.

Lab 3: Advanced Scheduling Simulation

Simulate **EDF (Earliest Deadline First)** and **Rate Monotonic (RM)** scheduling using SimSo.

Lab 4: Resource Sharing & Priority Inversion

Demonstration of **priority inversion**. Implement **priority inheritance** protocol.

Lab 5: Real-Time Communication (Mini Simulation)

Use NS-3 or socket programming to simulate soft vs. hard real-time message delivery.

Lab 6: RTOS Benchmarking & Case Study

Measure task latency and context switch time on RTLinux/FreeRTOS. Short case study of a real RTOS (RTLinux or FreeRTOS features).

Suggested Readings:

1. Liu, J. W. S. (2000). *Real-Time Systems*. Prentice Hall.
2. Mall, R. (2007). *Real-Time Systems: Theory and Practice*. Pearson Education.
3. Krishna, C. M., & Shin, K. G. (2010). *Real-Time Systems*. McGraw-Hill.
4. Laplante, P. A., & Ovaska, S. J. (2012). *Real-Time Systems Design and Analysis: Tools for the Practitioner*. Wiley.
5. Laplante, P. A. (2004). *Real-Time Systems Design and Analysis* (3rd ed.). Wiley-IEEE Press.

PROJECT / DISSERTATION (RD)

[to be undertaken by students opting for and eligible for Hons. with Research]

DISSERTATION:
Credit: 8; Full Marks: 150

Guidelines:

- Students will prepare a **dissertation of 10,000 words (approx.)** with a suitable title
- **Writing the Dissertation**:
  Introducing the area of research, Literature Review, Locating research gaps, Spelling out the central argument/hypothesis, Preparing the theoretical framework, Making the core chapters, Conclusion and Scope for further research, Endnotes to individual chapters, List of Works Cited/Bibliography, Appendix, if any.

Structure of the dissertation

Preliminary Pages

- Title Page: Includes name of the student, university, degree, and submission date.
- Certificate/Declaration: Signed statement of originality and supervisor's approval
- Abstract: A 250-350 word summary covering research problem, methodology, and core findings.
- Acknowledgements: To supervisors and others, if any.
- Table of Contents: Clear hierarchical mapping of chapters with page number
- List of Abbreviations: Specialized terms, if any

Introduction

- Background & Context: Situates the topic in historical or theoretical context.
- Research Problem & Objectives: The specific question or gap the research is addressing.
- Thesis Statement/Hypothesis: The core argument the dissertation intends to prove.
- Scope and Significance: Why the study matters.
- Literature Review & Theoretical Framework

Literature Review: Analyzes existing scholarly works to show the gap the study fills.

- Theoretical Framework: Identifies the philosophers, critics, or schools of thought shaping the analysis in the dissertation.

Methodology

- Describes *how* the research was conducted - Anyone or a combination of textual analysis/ archival research/ conducting interviews/ ethnographic/semi-ethno-graphic research involving both textual analysis and field visit etc.
- Thematic or Analytical Chapters

The Core Body: Usually 3 to 4 chapters focusing on the core concerns of the study.

- Structure: Each chapter introduces its specific theme, analyzes primary and secondary sources, and contributes to the overarching thesis.

Conclusion

- Summary & Synthesis: Recaps your main points without simply repeating them.

Final Verdict: Answers the research questions posed in the introduction.

Future Scope: Suggests how future scholars can build upon your work.

- Works Cited/ Reference, and/or Bibliography: A comprehensive list of all primary and secondary sources cited as per the latest MLA style.

Appendices: Transcripts, interview logs, or raw archival data, if any.

Marks distribution-

Sl. No	Particulars	Marks
1	Research Project / Dissertation Report	70
	▪ Proposal and Abstract (along with some keywords)	10
	▪ Literature Review &Theoretical Framework	10
	▪ Methodology and Research design	10
	▪ Thematic Analysis (Chapter-wise)	20
	▪ Report Quality	20
2	Seminar Presentation	30
3	Viva-Voce	50
Total		**150**

MINOR (MI)

Minor (MI)-08: Cloud Computing **Credits 04 (Full Marks: 75)**

Course Outcome:

Students able to:

CO1: Learn fundamental concepts of cloud infrastructure to evaluate trade-offs in power, efficiency, and cost, and manage data centers for resilient, elastic, and cost-efficient cloud applications.

CO2: Learn to describe system, network, and storage virtualization techniques and explain their role in enabling cloud computing.

CO3: Learn to illustrate the principles of cloud storage and demonstrate their implementation in systems such as Amazon S3 and HDFS.

CO4: Analyze various cloud programming models and apply them to develop and deploy solutions on the cloud.

MI-08T: Cloud Computing (Theory) *Credits 03 (45 hrs)*

Course Contents-

Unit 1: Introduction: **[3L]**

Essentials, Benefits and need for Cloud Computing - Business and IT Perspective - Cloud and Virtualization - Cloud Services Requirements - Cloud and Dynamic Infrastructure - Cloud Computing Characteristics Cloud Adoption.

Unit 2: Cloud Models: **[4L]**

Cloud Characteristics - Measured Service - Cloud Models - Security in a Public Cloud Public versus Private Clouds - Cloud Infrastructure Self Service

Unit 3: Cloud as a Service: **[3L]**

Gamut of Cloud Solutions - Principal Technologies - Cloud Strategy Cloud Design and Implementation using SOA - Conceptual Cloud Model - Cloud Service Defined

Unit 4: Cloud Solutions: **[3L]**

Cloud Ecosystem - Cloud Business Process Management - Cloud Service Management - Cloud Stack - Computing on Demand (CoD) – Cloud sourcing.

Unit 5: Cloud Offerings: **[3L]**

Information Storage, Retrieval, Archive and Protection - Cloud Analytics Testing under Cloud - Information Security - Virtual Desktop Infrastructure - Storage Cloud.

Unit 6: Cloud Management: **[3L]**

Resiliency – Provisioning - Asset Management - Cloud Governance - High Availability and Disaster Recovery - Charging Models, Usage Reporting, Billing and Metering.

Unit 7: Cloud Virtualization Technology: **[5L]**

Virtualization Defined - Virtualization Benefits - Server Virtualization - Virtualization for x86 Architecture - Hypervisor Management Software - Logical Partitioning (LPAR) - VIO Server - Virtual Infrastructure Requirements.

Unit 8: Cloud Storage Virtualization: **[5L]**

Storage virtualization - Storage Area Networks - Network-Attached storage - Cloud Server Virtualization - Virtualized Data Center.

Unit 9: Cloud and SOA: **[3L]**

SOA Journey to Infrastructure - SOA and Cloud - SOA Defined - SOA and IaaS - SOA-based Cloud Infrastructure Steps - SOA Business and IT Services.

Unit 10: Cloud Infrastructure Benchmarking: **[3L]**

OLTP Benchmark - Business Intelligence Benchmark - e-Business Benchmark - ISV Benchmarks - Cloud Performance Data Collection and Performance Monitoring Commands - Benchmark Tools.

MI-08P: Cloud Computing (Lab) *Credits 01 (30 hrs)*

List of Program:

1) Install Virtualbox/VMware Workstation with different flavours of linux or windows OS on top of windows7 or 8.
2) Install a C compiler in the virtual machine created using virtual box and execute Simple Programs
3) Install Google App Engine. Create ―hello world app‖ and other simple web applications using python/java.
4) Use GAE launcher to launch the web applications.
5) Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim.
6) Find a procedure to transfer the files from one virtual machine to another virtual machine.
7) Find a procedure to launch virtual machine using trystack (Online Openstack Demo Version)
8) Install Hadoop single node cluster and run simple applications like wordcount.

Suggested Readings:

1. Buyya, R., Broberg, J., & Goscinski, A. M. (Eds.). (2011). *Cloud Computing: Principles and Paradigms*. John Wiley & Sons.
2. Hurwitz, J., Bloor, R., Kaufman, M., & Halper, F. (2010). *Cloud Computing For Dummies*. Wiley Publishing

******