VIVEKANAND COLLEGE, KOLHAPUR

(EMPOWERED AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE

Two Years PG Programme

Department/Subject Specific Core or Major (DSC)

Curriculum, Teaching and Evaluation Structure

for

M.Sc. I Computer Science

Semester: I & II

(Implemented from academic year 2023-24 onwards)

VIVEKANAND COLLEGE, KOLHAPUR (EMPOWERED AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE

Syllabus for the Master of Science in Computer Science
M. Sc. I (Sem. I & II)
(National Education Policy 2020)
Applicable From Academic Year: 2023 – 2024

1. **Title**: M. Sc. Computer Science,
   Vivekanand College, Kolhapur (Empowered Autonomous)
2. **Faculty**: Faculty of Science and Technology.
3. **Year of Implementation**: For M. Sc. I (Semester I and Semester II): From July 2023 and for M. Sc. II (Semester III and Semester IV): From July 2024.
4. **Programme Outcomes (POs):**

   After completing the M. Sc. Programme, the students will able to:

   PO1: Demonstrate and apply the fundamental knowledge of the basic principles of sciences in various fields.
   PO2: Create awareness and a sense of responsibility towards the environment and society to solve the issues related to environmental pollution.
   PO3: Apply their professional, social, and personal knowledge.
   PO3: Competent to pursue research or pursue a career in the subject.
   PO4: Apply knowledge to build up small-scale industries for developing endogenous products.
   PO5: Communicate scientific information in a clear and concise manner both orally and in writing.
   PO6: Inculcate logical thinking to address a problem and become result oriented with a positive attitude.

5. **Programme Specific Outcomes (POs):**

   After completing the M. Sc. (Computer Science) Programme, the students will able to:

   PSO1: To analyze complex computational problems, develop innovative solutions, and apply mathematical and algorithmic principles to address a wide range of real-world challenges.
   PSO2: Proficient in multiple programming languages and software development methodologies, enabling them to design, develop, and maintain software systems of varying complexity.
   PSO3: Equipped with the skills necessary to engage in cutting-edge research in computer science, including the ability to identify research problems, conduct experiments, and contribute to the development of new knowledge in the field.
   PSO4: Proficient in data analysis techniques and machine learning algorithms, allowing them to extract valuable insights from data, build predictive models, and apply machine learning to a variety of domains.
   PSO5: understand the principles of cybersecurity and information assurance, enabling them to assess and mitigate security risks, protect data and systems, and contribute to the development of secure software and network infrastructures.
   PSO6: ability to collaborate effectively in multidisciplinary teams, communicate technical information clearly to both technical and non-technical audiences, and adapt to changing technologies and work environments.

Teaching and Evaluation Scheme

DEPARTMENT OF COMPUTER SCIENCE

M. Sc. I (Sem. I & II) From Academic Year: 2023 – 2024

Sr. No.	Course Abbr.	Course code	Course Name	Teaching Scheme Hours/week		Examination Scheme and Marks				Course Credits
				TH	PR	ESE	CIE	PR	Marks	
Semester-I										
1	DSC-I	DSC19CSC11	Problem Solving using Python	4	-	80	20	-	100	4
2	DSC-II	DSC19CSC12	Advanced Database Techniques	4	-	80	20	-	100	4
3	DSE-I	DSE19CSC11	Design and Analysis of Algorithms	4	-	80	20	-	100	4
		DSE19CSC12	Theory of Computer Science		-			-		
		DSE19CSC13	Subject relevant MOOC (NPTEL)		-			-		
		DSE19CSC14	Subject relevant MOOC (Swayam)		-			-		
4	RMD-I	RMD19CSC11	Research Methodology	4	-	80	20	-	100	4
5	DSC-PR-I	DSC19CSC19	Computer Science Lab-I	-	8		-	100	100	4
6	DSC-PR-II	DSC19CSC19	Computer Science Lab-II	-	4		-	50	50	2
				16	**12**	**320**	**80**	**150**	**550**	**22**
Semester-II										
1	DSC-III	DSC19CSC21	Advanced Java Programming	4	-	80	20	-	100	4
2	DSC-IV	DSC19CSC22	Digital Image Processing	4	-	80	20	-	100	4
3	DSE-II	DSE19CSC21	Data Mining and Data Warehousing	4	-	80	20	-	100	4
		DSE19CSC22	Embedded and IoT Technology		-			-		
		DSE19CSC23	Subject relevant MOOC (NPTEL)		-			-		
		DSE19CSC24	Subject relevant MOOC (Swayam)		-			-		
4	DSC-PR-III	DSC19CSC29	Computer Science Lab-III	-	8	-	-	100	100	4
5	DSC-PR-IV	DSC19CSC29	Computer Science Lab-IV	-	4	-	-	50	50	2
6	OJT-I	OJT19CSC21	On Job Training		4	-	-	-	100	4
				12	**16**	**240**	**60**	**150**	**550**	**22**
	Total (Sem. I & II)			**28**	**28**	**560**	**140**	**300**	**1100**	**44**

M. Sc. Computer Science Part – I: Semester – I

DSC-I (DSC19CSC11): Problem Solving using Python

Course Outcomes: Students will able to

CO1: Understand the fundamental algorithms and data structures, enabling them to analyze and solve complex problems efficiently using Python programming.

CO2: Acquire advanced proficiency in Python programming, including knowledge of its syntax, libraries, and best practices.

CO3: Apply their Python programming skills to solve real-world problems across various domains, including but not limited to data analysis, machine learning, and software development.

CO4: Develop skills in working with others to solve problems using Python.

UNIT-I	**Introduction to Programming Languages:** Programming languages, their classification and characteristics, language translators and language translation activities, Planning the Computer Program, What is program and programming paradigms, Concept of problem Solving, Problem definition, Program design, Debugging, Types of errors in programming, Documentation Techniques of Problem Solving: Algorithms, Flowcharting, Structured programming concepts, Programming methodologies viz. top-down and bottom-up programming	**20**
UNIT- II	**Building Blocks of Program:** Data, Data Types, Data Binding, Variables, Constants, Declaration, Operations on Data such as assignment, arithmetic, relational, logical or Boolean, ternary, bitwise, increment or decrement operators. Introduction to Python Programming: Features, Structure of a Python Program (Python Shell Indentations, Comments), Python Interpreter, Writing and executing simple program, Basic Data Types: numbers (int, long, float, complex), strings, Declaring variables, Performing assignments, arithmetic operations, Sequence Control – Precedence of operators, Type conversion, Simple input-output (print(), raw_input(), input())	**20**
UNIT- III	**Logics**: Conditional Statements: if, if-else, nested if –else Looping: for, while, nested loops, else clause with while and for loop Control statements: Terminating loops, skipping specific conditions (break, continue, pass) Numeric Functions: abs(), ceil(), floor(), max(), min(), pow(), sqrt() String Manipulation: Declaring strings, String immutability, unicode string (u'String'), escape sequences (\), Operations on String (Concatenation (+), Repetition (*), Slicing ([index]), Range Slicing([start:end] or [:end] or [start:] , Member ship operator (in, not in)), String Functions : capitalize(), len(), lower(), swapcase(), upper()	**20**

References:

1. Introduction to Python Programming and Data Structures, Danie Liang, Pearson, 2022
2. Charles Dierbach, Introduction to Computer Science using Python, Wiley, 2013
3. James Payne, Beginning Python: Using Python 2.6 and Python 3, Wiley India, 2010
4. Paul Gries, Jennifer Campbell, Jason Montojo, Practical Programming: An Introduction to Computer Science Using Python 3, Pragmatic Bookshelf, 2/E 2014.

DSC-II (DSC19CSC12): Advanced Database Techniques

Course Outcomes: Students will able to

CO1: Demonstrate proficiency in translating SQL queries into relational algebra and other operators.

CO2: Acquire a comprehensive understanding of transaction management, concurrency control and recovery techniques.

CO3: Understand the fundamentals of distributed databases and distributed database management systems (DDBMS).

CO4: Gain insights into NoSQL databases, including their features, advantages and disadvantages.

UNIT-I	**Query Processing and Optimization** Translating SQL Queries into Relational Algebra and Other Operators, Algorithms for External Sorting Algorithms for SELECT Operation, Implementing the JOIN Operation, Algorithms for PROJECT and Set Operations, Implementing Aggregate Operations and Different, Types of JOINs, Combining Operations Using Pipelining, Parallel Algorithms for Query Processing, Query Trees and Heuristics for Query Optimization, Choice of Query Execution Plans, Use of Selectivities in Cost-Based Optimization, Cost Functions for SELECT Operation, Cost Functions for the JOIN Operation, Overview of Query Optimization in MySQL	**20**
UNIT- II	**Transaction Management, Concurrency Control, and Recovery Techniques** The Concept of a Transaction, Transactions and Schedules, Concurrent Execution of Transactions, Two-Phase Locking Techniques for Concurrency Control, Concurrency Control Based on Timestamp Ordering, Multiversion Concurrency Control Techniques, Validation (Optimistic) Techniques and Snapshot Isolation Concurrency Control, Granularity of Data Items and Multiple Granularity Locking, Deadlock and Deadlock Handling-Deadlock Avoidance (wait-die, wound-wait), Deadlock Avoidance (Wound-Wait, Wait-die), Deadlock Detection and Recovery (Wait For Graph) Recovery Concepts, NO-UNDO/REDO Recovery Based on Deferred Update, Recovery Techniques Based on Immediate Update, Shadow Paging, The ARIES Recovery Algorithm, Recovery in Multidatabase Systems, Database Backup and Recovery from Catastrophic Failures	**20**
UNIT- III	**Introduction of Distributed Databases** Definition of Distributed databases and Distributed Database Management System (DDBMS), Promises of DDBMS, Complicating Factors, Problem Areas, DDBMS Architecture: DBMS standardization, Architectural models for DDBMS, Distributed database design: Design problem of distributed systems, Design, strategies (top-down, bottom-up), Fragmentation, Allocation and replication of fragments. **Introduction to NoSQL** Introduction to NoSQL Systems, Features of NoSQL, Advantage and Disadvantage of NOSQL, The CAP Theorem, Document-Based NoSQL Systems and MongoDB, NoSQL Key-Value Stores, Column-Based or Wide Column NoSQL Systems, NoSQL Graph Databases and Neo4j	**20**

References:

1. Database System Concepts by Sudarshan, Korth (McGraw-Hill Education)
2. An introduction to Database System – Bipin Desai, Galgotia Publications
3. Principles of Distributed Database Systems; 2nd Edition by M. Tamer Ozsu and Patrick Valduriez Publishers: Pearson Education Asia ISBN: 81-7808-375-2
4. Distributed Database; Principles & Systems by Publications, Stefano Ceri and Giuseppo Pelagatti, McGraw-Hill International Editions (1984)
5. MongoDB: The Definitive Guide, Second Edition, By Kristina Chodorow

DSE-I (DSE19CSC11): Design and Analysis of Algorithms

Course Outcomes: Students will able to

CO1: Analyze performance of algorithms, choose the appropriate data structure like Stack, Queue and Linked List, tree, graph and algorithm design method for a specified application.

CO2: Understand how the choice of data structures and algorithm design methods impacts the performance of programs.

CO3: Solve problems using algorithm design methods such as the greedy method, divide and conquer, dynamic programming, backtracking and branch and bound.

CO4: Demonstrate the ability to identify and solve real-world problems using algorithmic techniques learned in the course.

UNIT-I	**Algorithm Analysis:** Introduction to algorithms, characteristics of algorithm, designing algorithms, Algorithm design techniques, Analysis of an algorithm, asymptotic notations, Analysis of recursive algorithms. **Solving recurrence relation:** Substitution method, recursion tree method, master method. **Divide and Conquer approach:** Binary Search, Sorting (Merge Sort, Quick Sort). **Hashing:** Hashing, Hash tables, **Hash functions:** Division method, Mid-Square method, Folding method. **Collision and Collision resolution techniques:** Collision resolution by Open Addressing and Collision resolution by Chaining.	**20**
UNIT- II	**Data Structures:** Stacks, Queues, Linked list, Trees, Graph. **Trees:** Tree terminologies, **Tree traversal:** Inorder, Preorder and Postorder. **Binary tree:** Types of binary tree, **Binary search tree:** Operations on binary search tree. AVL tree, B-trees. **Graphs:** Introduction, Terminologies associated with graph. **Types of graph**, **Representations of graph:** Sequential representation and Linked List representation, **Traversing Graphs:** Breadth-first search, Depth-First Search. **Topological sort. Minimum Spanning trees:** Kruskal's algorithm, Prim's algorithm, **Shortest paths algorithm**	**20**
UNIT- III	**Greedy Algorithm:** General Characteristics of greedy algorithms, Elements of Greedy Strategy. **Problem solving using Greedy Algorithm:** Activity selection problem, Job Scheduling Problem, Huffman code. **Backtracking:** Introduction, N Queen Problem, Subset Sum and Hamiltonian Cycle. **Branch and Bound:** Introduction, 0/1 Knapsack, Travelling Salesman problem. Dynamic Programming: General method, applications-Matrix chain multiplication, optimal binary search trees.	**20**

References:

1. Design & Analysis of Algorithms 4Th Edition 2019 by Gajendra Sharma, Khanna Publishing House
2. Design And Analysis of Algorithms, Oxford University Press
3. James Payne, Beginning Python: Using Python 2.6 and Python 3, Wiley India, 2010
4. Paul Gries, Jennifer Campbell, Jason Montojo, Practical Programming: An Introduction to Computer Science Using Python 3, Pragmatic Bookshelf, 2/E 2014.

DSE-II (DSE19CSC12): Theory of Computer Science

Course Outcomes: Students will able to

CO1: Understand the basic mathematical concepts such as set theory, relations, functions, mathematical induction, and recursive definitions.

CO2: Understand the concepts of alphabets, strings, and languages, and be able to design regular grammars and finite automata.

CO3: Gain proficiency in computability theory, understanding the concepts of Turing machines, recursive and recursively enumerable languages, and the implications of undecidability and incompleteness theorems.

CO4: Apply theoretical concepts to practical scenarios in compiler design.

UNIT-I	**Mathematical Preliminaries and Automata Theory** Set theory, relations, functions, Mathematical induction and recursive definitions, Propositional and predicate logic, Mathematical proof techniques, Formal Languages and Automata: Alphabets, strings, languages, Regular languages and regular expressions, Finite automata and their types, Regular grammars and their equivalence to regular languages, Context-free languages, context-free grammars, and pushdown automata, Properties of Regular and Context-Free Languages: Closure properties, Decision properties, Pumping lemma for regular and context-free languages, Closure under homomorphisms, union, intersection, concatenation, Kleene star.	**20**
UNIT- II	**Computability Theory** Turing Machines: Definition and representation, Computability and decidability, Church-Turing thesis, Variants of Turing machines, Recursive and Recursively Enumerable Languages: Recursive and recursively enumerable sets, Universal Turing machine, The Halting problem, Undecidability and Incompleteness: Undecidable problems, Diagonalization, Gödel's incompleteness theorems **Complexity Theory:** Time and Space Complexity: Time and space complexity classes, Polynomial-time reductions, Cook-Levin theorem and NP-completeness, Complexity Classes: P and NP classes, NP-hardness, PSPACE and other complexity classes, Approximation Algorithms: Definition and examples, Polynomial-time approximation schemes (PTAS)	**20**
UNIT- III	**Formal Languages and Compiler Design** Chomsky Hierarchy and Parsing Techniques: Chomsky hierarchy of languages, Context-sensitive grammars and languages, Parsing techniques: LL and LR parsers, Compiler Design: Lexical analysis and regular expressions, Syntax analysis and parsing techniques, Semantic analysis and optimization, Code generation and code optimization Automated Theorem Proving: Resolution method, Model-checking algorithms, Applications in artificial intelligence and formal verification	**20**

References:

1. John C Martin, “Introduction to Languages and the Theory of Computation”, Tata McGraw Hill, 2009.
2. John E Hopcroft, Jeffrey D Ullman and Rajeev Motwani, "Introduction to Automata Theory, Languages and Computation," Pearson Education, 2009.
3. Theory of Computer Science, K. L. P Mishra, Chandrasekharan, PHI,3rd Edition
4. Introduction to Computer Theory, Daniel Cohen, Wiley,2nd Edition

RMD-I (RMD19CSC11): Research Methodology

Course Outcomes: Students will able to

CO1: Understand the principles and significance of research methodology in computer science.

CO2: Identify and formulate research problems, defining clear research questions and hypotheses.

CO3: Apply both quantitative and qualitative research methods, including data collection and analysis techniques, to address research questions.

CO4: Demonstrate effective research communication skills

UNIT-I	**Fundamentals of Research Methodology:** Meaning, Objectives, Motivation and Types of Research, Research Approaches. Significance of Research, Research Methods versus Methodology, Research and Scientific Method, Importance of Knowing How Research is done? Criteria of Good Research, research process and steps involved Hypothesis: Meaning, function and types of hypotheses; Null/Alternative hypothesis, Literature survey, sources of information, review. Ethical issues and intellectual property rights. Publication process, selection of journals, citation index, impact factor, h-index, i10 index, Journal Cite Score, Google scholar index, Research gate, Academia, etc.	**15**
UNIT- II	**Interpretation and Report Writing** Meaning of Interpretation, Technique of Interpretation, Precaution in Interpretation. Significance of Report Writing, Different Steps in Writing Report, Layout of the Research Report, Types of Reports. Mechanics of Writing a Research Report: Writing preliminaries, main body of research, references and bibliography. Precautions for Writing Research Reports. Meaning and importance of workshop, seminar, conference, symposium, etc. in research. Plagiarism- Concept and significance of plagiarism. Writing tools: Grammerly, Answerthepublic, Quillbot, Notion, Buzzsumo, Copyscape, Chatgpt, ginger. Referencing and citation tools: Endnote, Mendeley, Jabref, Zotero.	**15**
UNIT- III	**Computer Applications** Database Search engines: Google Scholar, Researchgate, Wiki- Databases, Science Direct, SciFinder, Scopus, inflibinet, Cambridge structural database, Web of Science, Indian Citation Index. Publishers in Computer Science: IEEE, Taylor and Francis, Elsevier, Springer, Science Direct, Wiley. Computer based laboratory, statistics and data interpretation, Computer based information systems for e.g docking, Scilab. Softwares for Computer Science: Python, Numpy, Pandas, Google Colab, Matlab, SPSS, Latex.	**15**
UNIT- IV	**Research Methods in Computer Science** Research Areas: Artificial Intelligence and Machine Learning, Computer Networks and Distributed Systems, Cybersecurity and Privacy, Human-Computer Interaction (HCI), Data Science and Big Data, Internet of Things (IoT) and Embedded Systems, Emerging Technologies and Future Directions: Quantum Computing, Blockchain. Research Design, Data Collection: Quantitative and qualitative data, Data Analysis, Statistical analysis in computer science research, Qualitative data analysis methods, Data visualization techniques.	**15**

References:

1. Kothari, C. R., Research Methodology-Methods and Techniques, 2nd Ed., New Age International, New Delhi.
2. Ram Ahuja, "Research Methods", (2001), Rawat Publications, New Delhi.
3. Kumar R., Research Methodology - A Step-By-Step Guide for Beginners, Pearson Education, Delhi (2006).

DSCPR-I (DSC19CSC19): Computer Science Lab-I

Practical's based on DSC19CSC11:

1. **Flowcharting and Algorithm Design:**
   Create a flowchart and write a Python program to find the factorial of a given number.

2. **Understanding Data Types and Variables:**
   Write a Python program that declares variables of different data types and performs arithmetic operations on them.

3. **Conditional Statements:**
   Create a program that uses if-else statements to determine whether a given number is even or odd.

4. **Looping Constructs:**
   Develop a Python program that uses a while loop to find the sum of all numbers from 1 to 100.

5. **Numeric Functions:**
   Write a program that takes a floating-point number as input and uses abs() to output its absolute value.

6. **String Manipulation:**
   Implement a Python program that concatenates two strings and prints the result.

7. **Understanding Python Interpreter:**
   Create a program that prints the Python version and interpreter information.

8. **Program Design with Functions:**
   Define a function that takes two parameters (length and width) and calculates the area of a rectangle.

9. **Using Break and Continue:**
   Write a program that uses the 'break' statement to exit a loop when a specific condition is met.

10. **Structured Programming Concepts:**
    Design a Python program that uses a top-down approach to solve a complex problem, breaking it into smaller subproblems.

11. **String Functions:**
    Develop a program that takes a user input string and uses string functions to capitalize the first letter.

12. **Bottom-Up Programming:**
    Design a program by starting with smaller functions and building up to a complete solution.

13. **Type Conversion:**
    Write a program that takes a numeric input as a string and converts it to an integer for mathematical operations.

DSCPR-II (DSC19CSC19): Computer Science Lab-II

Practical's based on DSC19CSC12:

1. **Translating SQL Queries into Relational Algebra:**
   Write a MySQL query and then express it in relational algebra.

2. **Algorithms for External Sorting:**
   Design a program to implement external sorting using MySQL.

3. **Algorithms for SELECT Operation:**
   Develop a MySQL query with proper indexing and analyze its execution plan.

4. **Implementing the JOIN Operation:**
   Create a MySQL query involving multiple tables with JOIN operations.

5. **Algorithms for PROJECT and Set Operations:**
   Write a MySQL query that includes PROJECT and Set Operations (UNION, INTERSECT, EXCEPT).

6. **Implementing Aggregate Operations:**
   Design a program to calculate aggregate functions (SUM, AVG, COUNT) in MySQL.

7. **Different Types of JOINs:**
   Implement LEFT JOIN, RIGHT JOIN, INNER JOIN, and FULL OUTER JOIN in MySQL.

8. **Combining Operations Using Pipelining:**
   Write a series of MySQL queries and analyze the results to demonstrate pipelining.

9. **Parallel Algorithms for Query Processing:**
   Explore and implement parallel query execution using MySQL.

10. **Query Trees and Heuristics for Query Optimization:**
    Create a query tree for a complex SQL query and optimize it using heuristics.

11. **Choice of Query Execution Plans:**
    Compare and analyze different execution plans for a given MySQL query.

12. **Concurrency Control Based on Timestamp Ordering:**
    Design a program to implement concurrency control using timestamp ordering in MySQL.

13. **Deadlock Handling - Deadlock Avoidance:**
    Implement a MySQL transaction with deadlock avoidance using either wait-die or wound-wait.

14. **The ARIES Recovery Algorithm:**
    Develop a program to simulate the ARIES recovery algorithm in MySQL.

M. Sc. Computer Science Part – I: Semester – II

DSC-III (DSC19CSC21): Advanced Java Programming

Course Outcomes: Students will able to
CO1: Demonstrate proficiency in advanced Java concepts and language features.
CO2: Effectively use Java for database connectivity and advanced database interactions.
CO3: Develop dynamic and scalable web applications using Java technologies
CO4: Apply Java in real-world scenarios to solve complex programming challenges.

UNIT-I	**Advanced Java Concepts and Language Features** Introduction to Advanced Java, Overview of Advanced Java, Importance of Advanced Java in modern software development, Introduction to swing, event handling, Multithreading, Understanding multithreading and concurrency, Synchronization and thread safety, Java I/O and Serialization, File I/O operations in Java, Serialization and deserialization, Working with different streams in Java	**20**
UNIT- II	**Introduction to JDBC** Overview of JDBC architecture, JDBC drivers and database connections, Executing SQL Queries with JDBC,CRUD operations with JDBC, Batch processing and transaction management, Connection Pooling and Data Source, Implementing connection pooling, Using Data Source for efficient database connections, Advanced JDBC Features, Callable Statements and stored procedures, Result Set types and concurrency control, Handling large datasets and streaming, ORM (Object-Relational Mapping) with Hibernate, Introduction to Hibernate framework, Mapping Java objects to database tables, Hibernate Query Language (HQL) and Criteria API	**20**
UNIT- III	**Introduction to Servlets** Overview of Servlet technology, Servlet life cycle and request handling, Java Server Pages (JSP), JSP syntax and directives, JSP expressions, declarations, and actions, Servlets and JSP Configuration in Web Applications, Deployment descriptors and web.xml, Servlet and JSP configuration in a web project, Java EE Web Applications, Building robust web applications using Java EE, Integrating servlets and JSP in a web project, Session Management and Security, Managing user sessions in web applications Implementing security features in Java web applications	**20**

References:

1. "Effective Java" by Joshua Bloch
2. "Java Concurrency in Practice" by Brian Goetz, Tim Peierls, Joshua Bloch, Joseph Bowbeer, David Holmes, and Doug Lea
3. "Head First Servlets and JSP" by Kathy Sierra and Bert Bates
4. "Pro Java EE Spring Patterns: Best Practices and Design Strategies" by Dhrubojyoti Kayal and John Carnell

DSC-III (DSC19CSC22): Digital Image Processing

Course Outcomes: Students will able to

CO1: Understand the fundamentals of digital image processing

CO2: Apply image analysis techniques for solving real-world problems.

CO3: Explore advanced topics in digital image processing, including image restoration, wavelet transforms, and applications in medical imaging.

CO4: Utilize deep learning approaches for image analysis and processing

UNIT-I	**Basics of Digital Image Processing:** Overview of digital image processing, Image acquisition and representation, Image enhancement techniques, Image Transforms and Filtering **Image transforms (Fourier, DCT):** Image filtering in the spatial and frequency domains, Convolution and correlation, Image Compression **Lossless and lossy compression techniques:** Image compression standards (JPEG, PNG), Transform coding and quantization, Color Image Processing **Color models and representations:** Color image enhancement and correction, Color image compression	**20**
UNIT- II	**Image Segmentation:** Techniques for image segmentation, Region-based and boundary-based segmentation, Thresholding and clustering methods, Object Recognition and Classification **Feature extraction and representation:** Pattern recognition in images, Machine learning in image classification, Image Morphology **Dilation and erosion operations:** Structuring elements and morphological operations Applications of morphological operations, Computer Vision Applications **Introduction to computer vision:** Object tracking and detection, Image-based 3D reconstruction	**20**
UNIT- III	**Image Restoration:** Degradation models and restoration techniques, Blind and non-blind restoration, Wiener filtering and constrained least squares restoration, Wavelet Transform in Image Processing **Introduction to wavelet transforms:** Wavelet-based image compression, Wavelet-based image denoising, Medical Image Processing **Basics of medical imaging:** Image processing applications in medical diagnosis, Challenges and advancements in medical image processing, Deep Learning for Image Processing **Convolutional Neural Networks (CNNs) for image analysis:** Transfer learning and fine-tuning for image tasks, Applications of deep learning in image processing	**20**

References:

1. "Digital Image Processing" by Rafael C. Gonzalez and Richard E. Woods
2. "Image Processing, Analysis, and Machine Vision" by Milan Sonka, Vaclav Hlavac, and Roger Boyle
3. "Digital Image Processing Using MATLAB" by Rafael C. Gonzalez, Richard E. Woods, and Steven L. Eddins
4. "Computer Vision: Algorithms and Applications" by Richard Szeliski

DSE-III (DSE19CSC21): Data Mining and Data Warehousing

Course Outcomes: Students will able to

CO1: Understand the fundamental concepts of data mining and data warehousing.

CO2: Design and implement a data warehouse, including ETL processes and dimensional modeling.

CO3: Apply various data mining techniques and algorithms to solve real-world problems

CO4: Explore advanced topics and emerging trends in data mining, including big data integration, spatial and temporal data mining.

UNIT-I	**Introduction to Data Mining:** Overview of data mining concepts, Applications and challenges in data mining, Data mining process and methodologies **Introduction to data warehousing:** Architecture of data warehouses, ETL (Extract, Transform, Load) processes, Data Warehouse Design and Implementation **Dimensional modeling:** Star and snowflake schemas, Data warehouse implementation best practices, Data Preprocessing in Data Mining **Data cleaning, integration and transformation:** Handling missing values and outliers, Data reduction and discretization	**20**
UNIT- II	**Data Mining Techniques and Algorithms:** Classification and Regression, Decision trees and rule-based classifiers, Naive Bayes and k-Nearest Neighbors (k-NN), Regression analysis in data mining, Clustering and Association Rule Mining **K-means clustering and hierarchical clustering:** Apriori algorithm for association rule mining, Evaluating clustering and association results, Text and Web Mining, Techniques for mining unstructured data, Sentiment analysis and opinion mining, Mining data from the web, Data Mining Tools and Applications, **Overview of popular data mining tools (e.g., Weka, RapidMiner):** Case studies and applications of data mining in different domains	**20**
UNIT- III	**Advanced Topics in Data Mining and Emerging Trends:** Advanced Data Mining Techniques, Ensemble methods (bagging, boosting), Support Vector Machines (SVM), Deep learning for data mining, Spatial and Temporal Data Mining, Mining spatial data patterns, Time series analysis and temporal data mining Applications of spatial and temporal data mining **Data Warehousing and Big Data:** Integrating data warehousing with big data technologies, Challenges and opportunities in big data analytics, Real-time data warehousing, Ethical and Privacy Issues in Data Mining, Ethical considerations in data mining, Privacy-preserving data mining techniques, Regulatory compliance and data governance	**20**

References:

1. "Data Mining: Concepts and Techniques" by Jiawei Han, Micheline Kamber, and Jian Pei
2. "The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling" by Ralph Kimball and Margy Ross
3. "Introduction to Data Mining" by Pang-Ning Tan, Michael Steinbach, and Vipin Kumar
4. "Big Data: A Revolution That Will Transform How We Live, Work, and Think" by Viktor Mayer-Schönberger and Kenneth Cukier

DSE-III (DSE19CSC22): Embedded and IoT Technology

Course Outcomes: Students will able to

CO1: Understand the fundamentals of embedded systems.

CO2: Design and develop IoT devices and applications.

CO3: Explore advanced topics in embedded systems and IoT and wireless sensor networks.

CO4: Apply their knowledge to real-world projects, demonstrating the ability to plan, develop, and present embedded systems and IoT solutions.

UNIT-I	**Introduction to Embedded Systems:** Definition and characteristics of embedded systems, Components and architecture of embedded systems, Applications of embedded systems in various domains, Microcontroller and Microprocessor Basics **Overview of microcontrollers and microprocessors:** Architecture and instruction set of popular microcontrollers, Memory types and interfacing in embedded systems, Programming Embedded Systems **Embedded programming languages (C, C++):** Real-time operating systems (RTOS) for embedded systems, Developing and debugging embedded applications, Sensors and Actuators in Embedded Systems **Types of sensors and actuators** Interfacing sensors and actuators with microcontrollers, Signal conditioning and processing	**20**
UNIT- II	**Introduction to IoT:** Definition and key concepts of the Internet of Things, IoT architecture and components, IoT protocols and communication technologies, IoT Devices and Platforms **Designing and developing IoT devices:** IoT hardware platforms and development boards, IoT connectivity options (Wi-Fi, Bluetooth, Zigbee, etc.), IoT Data Management and Analytics **Data acquisition and storage in IoT:** Cloud computing for IoT applications, Data analytics and visualization in IoT, IoT Security and Privacy, Security challenges in IoT, Encryption and authentication in IoT, Privacy concerns and ethical considerations	**20**
UNIT- III	**Embedded Systems in Real-time Applications:** Real-time constraints and challenges, Real-time operating systems (RTOS) in embedded systems, Case studies in real-time embedded applications, Edge Computing and Fog Computing, Introduction to edge and fog computing, Decentralized data processing in IoT, Applications and benefits of edge and fog computing, Wireless Sensor Networks (WSN) **Basics of WSN and their applications:** Protocols for WSN communication, Energy-efficient techniques in WSN, Embedded Systems and IoT Project Development, Planning and executing embedded systems and IoT projects, Integration of hardware and software components, Project demonstration and documentation	**20**

References:

1. "Embedded Systems: Architecture, Programming and Design" by Raj Kamal
2. "IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things" by David Hanes and Gonzalo Salgueiro
3. "Building Internet of Things with the Arduino" by Charalampos Doukas
4. "Real-Time Systems Design and Analysis: An Engineer's Handbook" by Phillip A. Laplante

DSCPR-III (DSC19CSC29): Computer Science Lab-III

Practical's based on DSC19CSC21:

1. Write a Java program that demonstrates multithreading.

2. Write a program in Java that performs file I/O operations in Java and includes serialization and deserialization.

3. Write a Java program that connects to a database using JDBC and performs CRUD operations.

4. Implement connection pooling and use a data source for efficient database connections in a Java application.

5. Write a Java program that utilizes Callable Statements and stored procedures with JDBC.

6. Write a Java program that demonstrates Object-Relational Mapping (ORM) using Hibernate, mapping Java objects to database tables.

7. Write a Java program that utilizes HQL and Criteria API for querying databases using Hibernate.

8. Write a Java program to implement a servlet demonstrating its life cycle and handling HTTP requests.

9. Create a simple JSP page with expressions, declarations, and actions.

10. Configure servlets and JSP in a web project using deployment descriptors and web.xml.

11. Build a robust web application using Java EE, integrating servlets and JSP.

12. Write a Java web application managing user sessions and implementing basic security features.

Practical's based on DSE19CSC21:

1. Load a dataset in Weka that contains missing values and outliers. Implement data preprocessing techniques to handle missing values and outliers. Evaluate the impact of preprocessing on the dataset.

2. Load a dataset suitable for classification in Weka. Apply at least three different classification algorithms (e.g., Decision Trees, Naive Bayes, k-NN) and compare their performance using cross-validation. Discuss the results.

3. Load a dataset in Weka suitable for clustering. Apply both K-means clustering and hierarchical clustering algorithms. Evaluate and compare the results of the two clustering techniques. Discuss the characteristics of the identified clusters.

4. Load a dataset in Weka that is suitable for association rule mining. Apply the Apriori algorithm to discover association rules. Adjust parameters such as minimum support and confidence, and discuss the generated rules.

5. Utilize a dataset containing text data in Weka. Apply text mining techniques such as TF-IDF representation and sentiment analysis using Weka's capabilities. Evaluate the effectiveness of the applied techniques.

DSCPR-IV (DSC19CSC29): Computer Science Lab-IV

Practical's based on DSC19CSC22:

1. Write a program to read and display a digital image. Apply basic enhancement techniques such as brightness adjustment and contrast stretching.

2. Implement a program to perform image filtering using convolution in the spatial domain. Apply common filters such as smoothing and sharpening.

3. Create a program to compress an image using the JPEG compression technique. Analyze the compression ratio and image quality.

4. Develop a program to enhance the colors of an RGB image. Apply techniques such as histogram equalization for color enhancement.

5. Implement a program to compress a color image using the PNG compression standard. Compare the results with JPEG compression.

6. Write a program to perform image segmentation using thresholding. Experiment with different thresholding methods.

7. Create a program that uses feature extraction techniques for object recognition. Implement a simple classification algorithm.

8. Develop a program to perform dilation and erosion operations on binary images using different structuring elements.

9. Implement a program that uses morphological operations for tasks such as noise removal and object boundary detection.

10. Write a program to track the movement of an object in a sequence of images.

11. Develop a program that reconstructs a 3D scene from a set of 2D images using structure-from-motion techniques.

12. Implement a program to restore a degraded image using Wiener filtering. Experiment with different degradation models.

13. Create a program that uses wavelet transform for denoising images. Compare the results with traditional filtering techniques.

14. Develop a program that applies image processing techniques to a medical image for diagnostic purposes.

15. Implement a CNN for image classification using a popular deep learning library. Explore transfer learning and fine-tuning.