ಮಂಗಳೂರು ವಿಶ್ವವಿದ್ಯಾನಿಲಯ

MANGALORE UNIVERSITY

(Accredited by NAAC Grade)

ಕ್ರಮಾಂಕ/No. : MU/ACC/CR.20/2025-26/A2

ಕುಲಸಚಿವರಕಛೇರಿ
ಮಂಗಳಗಂಗೋತ್ರಿ – 574 199
Office of the Registrar
Mangalagangothri – 574 199
ದಿನಾಂಕ/Date:31.07.2025

NOTIFICATION

Sub: Revised syllabus of Master of Computer Application [MCA] Programme.

Ref: Academic Council approval vide agenda No.:ಎಸಿಸಿ:ಶೈ.ಮ.ಸಾ.ಸ.1:1 (2025-26) dtd 18.07.2025.

*****

The revised syllabus of Master of Computer Application [MCA] Programme which has been approved by the Academic Council at its meeting held on 18.07.2025 is hereby notified for implementation with effect from the academic year 2025-26 and onwards.

Copy of the Syllabus shall be downloaded from the University Website (www.mangaloreuniversity.ac.in)

REGISTRAR

To,

1. The Registrar (Evaluation), Mangalore University.
2. The Chairman, PG Board of Studies in Computer Science and Computer Applications, Dept. of Computer Science, Mangalore University.
3. The Chairman, Dept. of Computer Science, Mangalore University.
4. The Principals of the College concerned.
5. The Asst. Registrar (ACC), O/o the Registrar, Mangalore University.
6. The Director, DUIMS, Mangalore University – with a request to publish in the website.
7. Guard File.

MANGALORE UNIVERSITY

Credits Pattern, Scheme of Examination and Syllabus for Two Years Master of Computer Applications (MCA) Degree Programme

Choice Based Credit System (CBCS) (2025-26)

POST-GRADUATE DEPARTMENT OF STUDIES AND RESEARCH IN COMPUTER SCIENCE
MANGALORE UNIVERSITY, MANGALAGANGOTHRI, KONAJE – 574 199
JUNE - 2025

Credits Pattern, Scheme of Examination and Syllabus for Two Years Master of Computer Applications (MCA) Degree Programme (CBCS Semester Scheme)

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs):

PEO1: To provide core theoretical and practical knowledge in the domain of Computer Applications for leading successful career in academia, industries, pursuing higher studies or entrepreneurial endeavors.

PEO2: To develop the ability to critically think, analyze and make decisions for offering techno-commercially feasible and socially acceptable solutions to real life problems in the areas of computing.

PEO3: To imbibe life-long learning, professional and ethical attitude for embracing global challenges and make positive impact on *environment and society.

The Programme Learning Objectives are:

PLO1: Scientific knowledge: Apply the knowledge of mathematics, science, and engineering fundamentals to the solution of complex scientific/societal/engineering problems.

PLO2: Problem analysis and Solutions: Identify, formulate, research literature, and analyze complex scientific/societal/engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences. Design solutions for complex scientific/societal/engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PLO3: Conduct investigations of complex problems and communication: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions. Communicate effectively on complex scientific/societal/engineering activities with the scientific community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PLO4: Modern tools usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex scientific/engineering activities with an understanding of the limitations.

PLO5: Environment and sustainability: Understand the impact of the professional scientific solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PLO6: Ethics and Team Work: Apply ethical principles and commit to professional ethics and responsibilities and norms of the social/scientific practice. Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PLO7: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PLO8: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAMME SPECIFIC OUTCOMES (PSOs):

PSO1: To identify, critically analyze, formulate and develop computer applications by applying knowledge of mathematics, computer science and management in practice.

PSO2: An ability to select modern computing tools and techniques and use them with dexterity and hence to design a computing system to meet desired needs within realistic constraints such as safety, security and applicability.

PSO3: An ability to devise and conduct experiments, interpret data and provide well informed conclusions and hence to understand the impact of system solutions in a contemporary, global, economic, environmental, and societal context for sustainable development.

PSO4: An ability to function professionally with ethical responsibility as an individual as well as in multidisciplinary teams with a positive attitude.

PSO5: An ability to communicate effectively and an ability to appreciate the importance of goal setting and to recognize the need for life-long learning.

Credits Pattern and Scheme of Examination

I SEMESTER M.C.A.								
Course Code	Courses	Theory Hours/ Week	Practical Hrs./ Week	Duration of exams (Hours)	Marks & Credits			
					IA	Exam	Total	Credits
HARD CORE								
25MCAH101	Mathematical Foundations of Computer Science	4L	-	3	30	70	100	4
25MCAH102	Operating Systems	4L	-	3	30	70	100	4
25MCAH103	Object Oriented Programming	4L	-	3	30	70	100	4
25MCAH104	Advanced Data Structures and Algorithms	4L	-	3	30	70	100	4
SOFT CORE								
25MCAS101	. NET Technology	4L	-	3	30	70	100	4
25MCAS102	Python Programming	4L	-	3	30	70	100	4
BRIDGE COURSE*								
25MCAB101	Foundations of Information Technology	4L	-	3	30	70	100	-
PRACTICALS [Any two labs shall be selected]								
25MCAP101	Operating Systems Lab	-	6	3	30	70	100	3
25MCAP102	Advanced Data Structures and Algorithms Lab	-	6	3	30	70	100	3
25MCAP103	.NET Technology Lab	-	6	3	30	70	100	3
25MCAP104	Python Programming Lab	-	6	3	30	70	100	3
	TOTAL	20+4*	12	21+3*	210	490	700	26

*Bridge Course: 25MCAB101: Foundation of Information Technology is a non-credit Course to be offered only for non-computer science background students. However, such students have to obtain eligibility both in IA and Final Examination.

II SEMESTER M.C.A.								
Course Code	Courses	Theory Hours/ Week	Practical Hours/ Week	Duration of exams (Hours)	Marks & Credits			
					IA	Exam	Total	Credits
HARD CORE								
25MCAH201	Principles of Data Science	4L	-	3	30	70	100	4
25MCAH202	Software Engineering	4L	-	3	30	70	100	4
25MCAH203	Data Communications and Computer Networks	4L	-	3	30	70	100	4
SOFTCORE [Any One course shall be selected from the list of courses]								
25MCAS201	Android Programming	4L	-	3	30	70	100	4
25MCAS202	Wireless Sensor Networks	4L	-	3	30	70	100	4
ELECTIVE - I [Within the Department]								
#	Elective - I	3L	-	3	30	70	100	3
PRACTICALS [Two practical shall be selected from the list]								
25MCAP201	Principles of Data Science Lab	-	6	3	30	70	100	3
25MCAP202	Data Communications and Computer Networks Lab	-	6	3	30	70	100	3
25MCAP203	Android Programming Lab	-	6	3	30	70	100	3
25MCAP204	Wireless Sensor Networks Lab	-	6	3	30	70	100	3
25MCAP205	Advanced Java Programming Lab	-	6	3	30	70	100	3
	TOTAL	19	12	21	210	490	700	25

BASED ON THE SELECTED ELECTIVE COURSE

SECOND SEMESTER ELECTIVE COURSES: ELECTIVE – I

Subject Code	Name of the Elective Course
25MCAE201	DISTRIBUTED COMPUTING
25MCAE202	ADVANCED JAVA PROGRAMMING
25MCAE203	OBJECT ORIENTED DATA MODELING
25MCAE204	CLOUD COMPUTING

III SEMESTER M.C.A.								
Course Code	Courses	Theory Hours/ Week	Practical Hours/ Week	Duration of exams (Hours)	Marks & Credits			
					IA	Exam	Total	Credits
HARD CORE								
25MCAH301	Machine Learning	4L	-	3	30	70	100	4
25MCAH302	Internet of Things	4L	-	3	30	70	100	4
25MCAH303	Advanced Database Management Systems	4L	-	3	30	70	100	4
SOFT CORE [Only ONE course shall be selected from the list of courses]								
25MCAS301	Digital Image Processing	4L	-	3	30	70	100	4
25MCAS302	Principles of Cyber Security	4L	-	3	30	70	100	4
ELECTIVE - II [Within the Department]								
#	Elective – II	3L	-	3	30	70	100	3
PRACTICALS [One practical shall be selected from the list]								
25MCAP301	Machine Learning Lab	-	6	3	30	70	100	3
25MCAP302	Internet of Things Lab	-	6	3	30	70	100	3
25MCAP303	Advanced Database Management Systems Lab	-	6	3	30	70	100	3
25MCAP304	Digital Image Processing Lab	-	6	3	30	70	100	3
25MCAP305	Natural Language Processing Lab	-	6	3	30	70	100	3
25MCAM301	Mini Project and Domain Knowledge Seminar	-	6	3	30	70*	100	3
Total		19	12	21	210	490	700	25

* The conduction of examination is similar to the practical examination which is evaluated based on the Mini Project Work.

Based on the Selected Elective Course

THIRD SEMESTER ELECTIVE COURSES: ELECTIVE - II

Course Code	Name of the Elective Course
25MCAE301	MOBILE COMPUTING
25MCAE302	SOFTWARE QUALITY ASSURANCE
25MCAE303	BLOCK CHAIN MANAGEMENT
25MCAE304	NATURAL LANGUAGE PROCESSING

IV SEMESTER MCA							
Course Code	Course	Practical Hours/ Week	Duration of Exam (Hrs)	Marks & Credits			
				IA	Dissertation + Viva Exam	Total	Credits
25MCAP401	Dissertation and Viva-Voce	24	—	100	200 (Dissertation Report : 100 Viva-Voce: 100)	300	12
25MCAE402	Course offered through MOOCS-SWAYAM/NPTEL (Minimum of 8 weeks duration course)					100	02
25MCAE403	Course offered through MOOCS-SWAYAM/NPTEL (Minimum of 8 weeks duration course)					100	02
TOTAL MARKS OF FIRST SEMESTER						700	26
TOTAL MARKS OF SECOND SEMESTER						700	25
TOTAL MARKS OF THIRD SEMESTER						700	25
TOTAL MARKS OF FOURTH SEMESTER						500	16
GRAND TOTAL CREDITS OF ALL THE FOUR SEMESTERS						**2600**	**92**

Note:

The dissertation work shall be carried out either in the University, Software Company, R&D Organization or any Institutes of National Importance.

***** The MOOCS-SWAYAM/NPTEL course shall be taken by students in online mode offered by SWAYAM/NPTEL course coordinator. The courses can be taken anytime during their Masters' programme and the courses completion certificate shall be obtained before the completion of fourth semester term days.

List of Hard Core, Soft Core and Elective Courses

Hard Core Courses			
Sl. No.	Course Code	Course Title	Total Credits
1.	25MCAH101	Mathematical Foundation of Computer Science	4
2.	25MCAH102	Operating Systems	4
3.	25MCAH103	Object Oriented Programming	4
4.	25MCAH104	Advanced Data Structure and Algorithms	4
5.	25MCAH201	Principles of Data Science	4
6.	25MCAH202	Software Engineering	4
7.	25MCAH203	Data Communications and Computer Networks	4
8.	25MCAH301	Machine Learning	4
9.	25MCAH302	Internet of Things	4
10	25MCAH303	Advanced Database Management Systems	4
11	25MCAP401	Dissertation with viva voce examination	12
12	25MCAE402	MOOCS Course-2	2
13	25MCAE403	MOOCS Course-2	2
TOTAL			56

Soft Core Courses			
Sl. No.	Course Code	Course Title	Total Credits
1	25MCAS101	. NET Technology	4
2	25MCAS102	Python Programming	
3	25MCAB101	Foundations of Information Technology	0
4	25MCAP101	Operating Systems Lab	3+ 3
5	25MCAP102	Advanced Data Structure and Algorithms Lab	
6	25MCAP103	. Net Technology Lab	
7	25MCAP104	Python Programming Lab	
8	25MCAS201	Android Programming	4
9	25MCAS202	Wireless Sensor Networks	
10	25MCAP201	Principles of Data Science Lab	3 + 3
11	25MCAP202	Data Communications and Computer Networks Lab	
12	25MCAP203	Android Programming Lab	
13	25MCAP204	Wireless Sensor Networks Lab	
14	25MCAP205	Advanced Java Programming Lab	
17	25MCAS301	Digital Image Processing	4
18	25MCAS302	Principles of Cyber Security	
19	25MCAP301	Machine Learning Lab	3
20	25MCAP302	Internet of Things Lab	
21	25MCAP303	Advanced Database Management Systems Lab	
22	25MCAP304	Digital Image Processing Lab	
23	25MCAP305	Natural Language Processing Lab	
24	25MCAM301	Mini Project and Domain Knowledge Seminar	3
Total			30

Elective Courses			
Sl. No.	Course Code	Course Title	Total Credits
1	25MCAE201	Distributed Computing	3
2	25MCAE202	Advanced Java Programming	
3	25MCAE203	Object Oriented Data Modeling	
4	25MCAE204	Cloud Computing	
6	25MCAE301	Mobile Computing	3
7	25MCAE302	Software Quality Assurance	
8	25MCAE303	Block Chain Management	
9	25MCAE304	Natural Language Processing	
Total			6

Percentage coverage of Hard core/Soft core/Elective Courses:

Hard Core Credits:	16 + 12+12+16	= 56	(60.87%)
Soft Core Credits:	10 +10+10	= 30	(32.60%)
Elective Credits:	03 +03	= 06	(6.52%)

QUESTION PAPER PATTERN

Name of the Examination: ____

Duration: 3 Hrs. Max. Marks: 70

Section – A: Answer all the questions and each question carries TWO marks

1. i)

   ii)

   iii)

   iv)

   v)

Section – B: Answer all the questions. Each question carries FIFTEEN marks

UNIT-1:	2. **OR** 3.
UNIT-2:	4. **OR** 5.
UNIT-3:	6. **OR** 7.
UNIT-4:	8. **OR** 9.

Note: *Each question of an UNIT under Section-B may contain multiple sub-questions, summing to fifteen marks.*

25MCAH101: MATHEMATICAL FOUNDATIONS OF COMPUTER SCIENCE

Hours/Week: 4 I.A. Marks: 30
Credits: 4 Exam. Marks: 70

Course Learning Objectives:

- The primary objective of this course is to provide mathematical background and sufficient experience on various topics of discrete mathematics like logic and proofs, combinatory, graphs, algebraic structures, formal languages and finite state automata.
- Course will extend student's Logical and Mathematical maturity and ability to deal with abstraction and to introduce most of the basic terminologies used in computer science courses and application of ideas to solve practical problems.
- On completion of this course, students should be able to demonstrate their understanding of and apply methods of discrete mathematics in CS to subsequent courses in algorithm design and analysis, automata theory and computability, information systems, computer networks.
- In particular, students should be able to - use logical notation to define fundamental mathematical concepts such as sets, relations, functions and various algebraic structures, reason mathematically using such structures, and evaluate arguments that use such structures.

Course Outcomes: After completing the course, the students will be able to,

- Understand basic probability axioms and rules and the moments of discrete and continuous random variables as well as be familiar with common named discrete and continuous random variables.
- Gain the knowledge of computing and mathematics appropriate to the discipline.
- Know the significance of mathematical foundations, algorithmic principles and computer science.
- Modeling and design of computer based systems in a way that demonstrates.
- Understand the design and development principles in the construction of software systems of varying complexity.
- Write and evaluate a proof or outline the basic structure of and give examples of each proof technique described.
- Recognize Model problems in Computer Science using graphs and trees.

UNIT-I **12 Hrs.**

Sets: Elements of a set, methods of describing a set, Tabular or Roster Method, Rule Method or Set Builder, Empty or Void or Null Set, Types of sets : Finite sets and Infinite sets, singleton, equal sets, subsets, Proper Subset, Power Set, Universal Set, Venn Diagrams, Operations on Sets, Union, Intersection of Sets, Disjoint Sets, Difference of two Sets, Symmetric Difference of Sets, Complement of a Set, De- Mogran's laws, Algebra of sets.
Relations: Introduction, Properties of a binary relation in a set, Relation matrix and graph of a relation, Equivalence relations, compatibility relations, composition of Binary relations.
Functions: Definition, graph of a function, types of functions: Surjective, bijective, Injective, Composition of functions, Inverse functions.
Introduction to Probability Theory: Definitions of Sample Space, Random Variables, Probability Distributions, Expected Values, Joint Distributions, Variance, Covariance related problems, Bayes' theorem statement and problems.

UNIT-II **12 Hrs.**

Propositional Logic: Introduction, Statements and Notation, Connectives-Negation, Conjunction, Disjunction, Statement, Formulas and Truth Tables, Conditional and Biconditional, Tautologies, contradiction, contingency, Equivalence of Formulas, Duality Law.

Predicate Logic: Limitations of Predicate Logic, Universal and Existential Quantification; Modus Ponens and Modus Tollens.

Proof Techniques: Notions of Implication, Converse, Inverse, Contrapositive, Negation, and Contradiction.

The Structure of Formal Proofs: Direct Proofs; Proof By Counter Example; Proof By Contraposition; Proof By Contradiction;

Mathematical Induction, Strong Induction; Recursive Mathematical Definitions.

UNIT-III **12 Hrs.**

Theory of Computation: Introduction, Strings and their properties, Formal Languages, Types of Grammars and Languages, Chomsky classification of Languages, Recursive And Recursively Enumerable Sets, Operations, Theory of Automata: Finite State Models, Minimization, Regular sets and Regular Grammars, Pumping Lemma, Closure properties, Applications of Finite automata.

UNIT-IV **12 Hrs.**

Context Free Languages: Context Free Grammar and Push Down Automata, equivalence of PDA and CFG, Deterministic PDA, Normal forms, Applications of CFG. Turing Machines: Representation and Design of TM, Halting problem, Universal TM and modifications.

REFERENCE BOOKS:

1. JD Ullman et al., Introduction to Automata Theory, Languages and Computation, 3rd Edition, Pearson Publication, 2006.
2. C L Liu, Elements of Discrete Mathematics: A Computer Oriented Approach, McGraw- Hill edition, 2013.
3. K. S. Trivedi, Probability and Statistics with Reliability, Queuing and Computer Science Applications, First Edition, Prentice Hall of India. 2008.
4. Schöning, Uwe, Pruim, Randall J, Gems of Theoretical Computer Science, Springer Publications.
5. Hary R Lewis, Christor H Papadimetrion, Elements of the Theory of Computation, Prentice-Hall International, 1998.
6. K L P Mishra and N Chandrashekar, Theory of Computer Science, 3rd Edition, PHI publication, 2007.

25MCAH102: OPERATING SYSTEMS

Hours/Week: 4 I.A. Marks: 30
Credits: 4 Exam. Marks: 70

Course Learning Objectives:

- Explore the structure of OS and basic architectural components involved in OS design.
- Analyze and design the applications to run in parallel either using process or thread models of different OS.
- Study the various device and resource management techniques for timesharing and distributed systems.
- Understand the Mutual exclusion, Deadlock detection and agreement protocols of Distributed operating system. Interpret the mechanisms adopted for file sharing in distributed Applications.

Course Outcomes: After completing the course, the students will be able to,

- Understand the structure of OS and basic architectural components involved in OS design.
- Analyze and design the applications to run in parallel either using process or thread models of different OS.
- Study the various device and resource management techniques for time sharing and distributed systems.
- Recognize the Mutual exclusion, Deadlock detection and agreement protocols of Distributed operating system.
- Interpret the mechanisms adopted for file sharing in distributed Applications.
- Evaluate the requirement for process synchronization and coordination handled by OS.
- Collecting and understanding the various security aspects of operating system.

UNIT-I 12 Hrs.

Operating System Overview : Operating System Objectives and Functions, The Evolution of Operating Systems, Major Achievements, Developments Leading to Modern Operating Systems, Microsoft Windows Overview, Traditional UNIX Systems, Modern UNIX Systems, Linux. Process Description & Control, Process Management: Process States, Process Description, Process Control, Process Synchronization, The Critical Section Problem, Peterson's Problem, Semaphores, Classic Problems of Synchronization.

UNIT-II 12 Hrs.

CPU Scheduling: Scheduling Criteria, Scheduling Algorithms, Thread Scheduling, Multiprocessor Scheduling, Real-Time Scheduling, Linux Scheduling, Windows Vista Scheduling. Virtual Memory: Hardware and Control Structures, Operating System Software, UNIX and Solaris Memory Management, Linux Memory Management, Windows Memory Management.

UNIT-III 12 Hrs.

Threads, SMP, and Microkernel: Processes and Threads, Symmetric Multiprocessing (SMP), Microkernels, Windows Thread and SMP Management, Linux Process and Thread Management: Distributed Process Management: Process Migration, Distributed Global States, Distributed Mutual Exclusion, Distributed Deadlock. Distributed File Systems: Naming and Transparency, Remote File Access, Stateful verses Stateless Service, File Replication.

UNIT-IV **12 Hrs.**

Protection and Security: Goals of Protection, Access Matrix- Security Problem, Computer Security Classifications, User Authentication - Program Threats and Systems Threats, Securing Systems and Facilities. Characteristics of Real time OS, Scheduling, Deadline scheduling, Priority inversion, Mobile operating systems- Features of iOS and Android.

REFERENCE BOOKS:

1. William Stallings, Operating Systems: Internals and Design Principles, 6th Edition, Prentice Hall, 2013.
2. Gary Nutt, Operating Systems, 3rd Edition, Pearson, 2014.
3. Silberschatz, Galvin, Gagne, Operating System Concepts, 8th Edition, Wiley, 2008
4. Andrew S. Tanenbaum, Albert S. Woodhull, Operating Systems Design and Implementation, 3rd Edition, Prentice Hall, 2006.
5. Pradeep K Sinha, Distributed Operating Systems, PHI, 2010.

25MCAH103: OBJECT ORIENTED PROGRAMMING

Hours/Week: 4 I.A. Marks: 30
Credits: 4 Exam. Marks: 70

Course Learning Objectives:

- The model of object oriented programming: abstract data types, encapsulation, inheritance and polymorphism.
- Fundamental features of an object oriented language like Java: object classes and interfaces, exceptions and libraries of object collections.
- Discuss the statement of a business problem and from this determine suitable logic for solving the problem; then be able to proceed to code that logic as a program written in Java.
- How to test, document and prepare a professional looking package for each business project using javadoc.

Course Outcomes: After completing the course, the students will be able to,

- Understand object oriented software development concepts.
- Study the principles of inheritance and polymorphism; and demonstrates how they relate to the design of abstract classes.
- Understand the implementation of packages and interfaces.
- Realize an exception handling, event handling and multithreading.
- Design Graphical User Interface using applets and swing.
- Understanding the threading and multithreading and their corresponding classes.
- Realize the importance12Hrs of Lambda expressions in OOPs.

UNIT-I **12 Hrs.**

Object Oriented Programming Principles, Need for OOP Paradigm, Introduction to Java, Characteristics, Data Types, Variables, Arrays. Control Statements: Selection, Iteration, Jump Statements, Operators, String Handling, String Comparison, String Buffer. Introduction to Classes, Class Fundamentals, Constructor, Methods, Stack Class, Inheritance, Creating Multilevel Hierarchy, Method Over-Riding

UNIT-II **12 Hrs.**

The Abstract Class; Packages, Defining a Package, CLASSPATH; Interface, Defining an Interface, Uses of Interfaces, Interfaces versus Abstract Classes. Exception Handling: Exception Classes; Common Exceptions; Exception Handling Techniques, Usage of try, catch, throw, throws and finally, built in exceptions, creating own exception sub classes.

UNIT-III **12 Hrs.**

Multi-threaded Programming: Introduction; Creating Threads: Extending Threads; Implementing Runnable; Synchronization, Priorities, Inter-Thread Communication, Thread States and Methods on Thread Objects. Event Handling: Two Event Handling Mechanisms; The Delegation Event Model; Event Classes; Sources of Events; Event Listener Interfaces; Using the Delegation Event Model; Adapter Classes; Inner Classes.

UNIT-IV **12 Hrs.**

Lambda Expressions: Introduction, Block Lambda Expressions, Generic Functional Interfaces, Passing Lambda Expressions as Arguments, Exceptions, Variable Capture, Method References, Constructor References, Predefined Functional Interfaces. Swing: Components and Containers; The Swing Packages; JLabel; ImageIcon; JTextField; The

Swing Buttons; Understanding Layout Managers; JTabbedPane; JScrollPane; JList; JComboBox; JTable; Overview of Menu.

REFERENCE BOOKS:

1. Herbert Schildt, Java the complete reference, 7th Edition, TMH.
2. T. Budd, Understanding OOP with Java, updated edition, Pearson Education.
3. J. Nino and F.A. Hosch, An Introduction to programming and OO design using Java, John Wiley & sons.
4. Y. Daniel Liang, Introduction to Java programming, Pearson Education.
5. R.A. Johnson, An introduction to Java programming and Object Oriented Application Development, Thomson.

25MCAH104: ADVANCED DATA STRUCTURES AND ALGORITHMS

Hours/Week: 4 I.A. Marks: 30
Credits: 4 Exam. Marks: 70

Course Learning Objectives:

- Understand and remember algorithms and its analysis procedure.
- Design and implement various data structures algorithms.
- To introduce various techniques for representation of the data in the real world
- Compute the complexity of various algorithms.

Course Outcomes: After completing the course, the students will be able to,

- Ensure that the student evolves as a competent programmer capable of design, analyze and implement algorithms and data structures for different kinds of problems.
- Expose the student to the algorithm analysis techniques, to the theory of reductions, and to the classification of problems into complexity classes like NP.
- Design and analyze programming problem statements, choose appropriate data structures and algorithms for a specific problem.
- Understand the necessary mathematical abstraction to solve problems, Come up with analysis of efficiency and proofs of correctness.
- Comprehend and select algorithm design approaches in a problem specific manner.
- Come across the importance of graphs and their features for the applications uses.
- Gathering the real strategies searching and sorting techniques.

UNIT-I **12 Hrs.**

Review of Basic Data Structures: Arrays, Stack, Queue, Circular Queue, Linked List-Singly Linked List, Doubly Linked List, Circular Linked List. Introduction to Algorithms: Algorithms, Performance Analysis – time complexity and space complexity, O-notation, Omega notation and Theta notation.

UNIT-II **12 Hrs.**

Introduction to Nonlinear Data Structures, Search Trees: Trees, Binary trees, Binary Tree Traversal, Applications of Binary Trees, Binary Search Trees- Searching, Insertion and Deletion on Binary Search Trees, Balanced Search Trees- AVL Trees- Insertion and deletion on AVL Trees, Red –Black Tress- Representation, Insertion and Deletion on Red –Black Trees, Splay Trees - Representation, Insertion and deletion on Splay Trees, Heaps: Representation, Insertion and Deletion on Heaps.

UNIT-III **12 Hrs.**

Graphs: Introduction to Graphs, digraphs, Sub-graphs, Paths, Walks, Graphs Representation, Graph Traversals - Depth-first and breadth-first traversal , Applications of graphs - Minimum Spanning Tree – Prim's and Kruskal's algorithms.
Hashing: Introduction to hashing, Hash Table Representation, Hash Functions, Collision Resolution-Separate Chaining, Open Addressing-Linear Probing, Quadratic Probing, Double Hashing.

UNIT-IV **12 Hrs.**

Design Strategies: Divide and Conquer- Binary Search, Merge Sort, Greedy method - Job sequencing with deadlines, Dynamic Programming – Optimal Binary Search Tree, Backtracking- 8 Queens problem, Introduction to NP-Hard and NP-Completeness.

REFERENCE BOOKS:

1. Mark A. Weiss, "Data structures and Algorithm analysis in C++ (Java)", Fourth Edition, PHI ,2013.
2. AnanyLevitin, "Introduction to the Design and Analysis of Algorithms" Pearson Education, 2015.
3. E. Horowitz, S.Sahni and Dinesh Mehta, "Fundamentals of Data structures in C++", University Press, 2007.

25MCAS101: .NET TECHNOLOGY

Hours/Week: 4 I.A. Marks: 30
Credits: 4 Exam. Marks: 70

Course Learning Objectives:

- The concept of .NET framework, building blocks of .NET framework and application development using IDE.
- C# programming language, use of windows forms and GUI based programs.
- OOP concepts, concept of assemblies and string manipulation.
- Designing of web applications and validating forms using validation controls, interacting with database using server side programming.

Course Outcomes: After completing the course, the students will be able to,

- Understand .NET framework, its runtime environment and application development using IDE of Visual Studio 2010 and higher versions.
- Develop well-defined programs using the C# programming language; learn to use Windows forms and to create GUI-based programs.
- Able to apply the principles of object-oriented programming and develop assemblies and deployment in .NET.
- Apply and build web applications and validation form data using validation controls.
- Create dynamic web applications that interact with a database using server-side programming.
- Understand Constructing classes, methods and instantiate objects.
- Understand and implement string manipulation, events and exception handling within .NET application environment.

UNIT-I **12 Hrs.**

Introduction: Principles of .NET, Overview of .NET Framework, Review of OOP Concepts – C# language fundamentals – Basic Elements of C# – Program Structure and simple Input and Output Operations – Data types –Value types –Reference types – Identifiers – Variables – Constraints –Literals – Operators and Expressions – Statements – Arrays and Structures. Object Oriented Programming Concepts: Encapsulation – Encapsulation Services – Pseudo- Encapsulation: Creating Read-Only Fields- Inheritance - Namespace – Polymorphism – Interface and Overloading – Multiple Inheritance – Property – Indexes – Delegates and Events – Publish/Subscribe Design Patterns- Operator Overloading– Method Overloading.

UNIT-II **12 Hrs.**

C# Concepts for creating Data Structures - File Operation – File Management systems – Stream Oriented Operations- Multitasking – Multithreading – Thread Operation – Synchronization– Exceptions and Object lifetime. Building C# Applications: The Role of the Command Line Complier – Building C # Applications, Working with csc.exe, Response Files– Generating Bug Reports – Remaining C# Compiler Options – The Command Line Debugger (cordbg.exe) – Using the Visual Studio .NET IDE – Other Key Aspects of the VS.NET IDE – C# "Preprocessor:" Directives.

UNIT-III **12 Hrs.**

.NET ASSEMBLERS and Windows Applications: An Overview of .NET Assembly – Building a Simple File Test Assembly– A C# Client Application– A Visual Basic .NET Client Application– Cross Language Inheritance– Exploring the CarLibrary's– Manifest– Exploring the CarLibrary's Types– Building the Multifile Assembly– Using Assembly–

Understanding Private Assemblies– Probing for Private Assemblies (The Basics) – Private Assemblies XML Configurations Files– Probing for Private Assemblies (The Details) – Understanding Shared Assembly – Understanding Shared Names– Building a Shared Assembly– Understanding Delay Signing– Installing/Removing Shared Assembly. Building Windows application –Working with c# controls– Event handling – Graphics Device Interface (GDI).

UNIT-IV **12 Hrs.**

ADO.NET and Database Connectivity: Introduction to ADO.NET– Major Components of ADO.NET– Establishing Database Connections– Connection objects– Command objects– Datasets– Data readers– Querying databases– Data Grid Views– Data Validation.

REFERENCE BOOKS:

1. Stephen C. Perry — "Core C# and .NET", Pearson Education, 2006.
2. S. ThamaraiSelvi and R. Murugesan —"A Textbook on C#" —, Pearson Education, 2003.
3. Andrew Troelsen, Pro C# with .NET 3.0 Special Edition, Dream tech Press, India, 2007.
4. E. Balagurusamy, Programming in C#, 5th Reprint, Tata McGraw Hill, 2004. (For Programming Examples)
5. Tom Archer, Inside C# WP Publishers, 2001.
6. Herbert Scheldt, C#: The Complete Reference, Tata McGraw Hill, 2004.
7. Robinson et al, -"Professional C#", Fifth Edition, Wrox Press, 2002.

25MCAS102: PYTHON PROGRAMMING

Hours/Week: 4 I.A. Marks: 30
Credits: 4 Exam. Marks: 70

Course Outcomes: After completing the course, the students will be able to,

- Explain basic principles of Python programming language
- Problem solving and programming capability.
- Implement object oriented concepts,
- Implement database and GUI applications.

Course Outcomes:

After completing the course, the students will be able to,

- Demonstrate their understanding of the fundamentals of python programming
- Show their skills of using python development framework
- Develop software with reasonable complexity and their design aspects.
- Deploy software to data analysis and debug the programs
- Understands the concept of data visualization for data analysis.
- Recognizes basics of SQLlite database and perform various possible operation on database.

UNIT-I **12 Hrs.**

Introduction to Python, the concept of data types; variables, assignments; immutable variables; numerical types; arithmetic operators and expressions; logical operators; ranges; Control statements: if-else, loops; String manipulations: subscript operator, indexing, slicing a string; strings and number System: converting strings to numbers and vice versa. Binary, octal, hexadecimal numbers.

UNIT-II **12 Hrs.**

Lists, tuples, and dictionaries; basic list operators, replacing, inserting, removing an element; searching and sorting lists; dictionary literals, adding and removing keys, accessing and Replacing values; traversing dictionaries; Design with functions: hiding redundancy, complexity; arguments and return values; formal vs. actual arguments, named arguments. Recursive functions; Python libraries to handle numerical data.

UNIT-III **12 Hrs.**

Classes and OOP: classes, objects, attributes and methods; defining classes; design with classes, inheritance, polymorphism, operator overloading, abstract classes; Modules and Packages; Manipulating files and directories, os and sys modules; text files: reading/writing text and Numbers from/to a file; creating and reading a formatted file (csv or tab-separated); Exception handling.

UNIT-IV **12 Hrs.**

Python database application programmer's interface (DB- API), connection and cursor objects, Type objects and constructors, python database adapters. Creating simple web clients, introduction to CGI, CGI module, building CGI applications, Python web application frameworks.

REFERENCE BOOKS:

1. Kenneth A. Lambert, The Fundamentals of Python: First Programs, 2011, Cengage Learning,
2. Magnus Lie Hetland, Beginning Python from Novice to Professional, Second Edition.
3. Mark Summerfield, Programming in Python 3 - A Complete Introduction to the Python Language, Second Edition.
4. Y. Daniel Liang, "Introduction to Programming Using Python", Pearson
5. Chun, J Wesley, Core Python Programming, 2nd Edition, Pearson, 2007 Reprint 2010.
6. David Beazley and Brian K. Jones, Python Cookbook, Third Edition, Shroff Publishers & Distributors Pvt. Ltd.
7. Mark Lutz, Learning Python FIFTH EDITION Mark Lutz.
8. Mark Lutz, Programming Python (English) 4th Edition.
9. Testing Python, David Sale, Wiley India (P) Ltd.,

25MCAB101: FOUNDATIONS OF INFORMATION TECHNOLOGY

Hours/Week: 4 I.A. Marks: 30
Credits: 0 Exam. Marks: 70

Course Learning Objectives:

- The basic concepts and terminology of Computer and information technology.
- Pursue specialized programs leading to technical and professional skills in computer programming.
- Skills relating to IT basics, computer applications, programming, interactive media.
- The basic data structures like array, linked lists etc. and their applications.

Course Outcomes: After completing the course, the students will be able to,

- Ensure that the student understand the Computer Fundamentals
- Understand the techniques of problem solving using programming language concepts
- Realize the importance of computer concepts and programming
- Recognize the basic data structures and their applications
- Gathering the importance of algorithms for scientific problems.
- Come across the flow design of a computer problem.
- Identical fundamentally realize the needs of the basic details.

UNIT-I **12 Hrs.**

Basics of Digital Computers and Digital Computing System: Number systems, Number base conversion, Complements, Binary codes, Binary arithmetic's. Boolean algebra: Definitions, Basic theorems and properties of Boolean algebra, Venn diagram. Fundamentals of Operating System.

UNIT-II **12 Hrs.**

Problem Solving Techniques: Introduction, Problem Solving Procedure. Algorithm: Steps involved in algorithm development, Algorithms for simple problems, Flowcharts, Pseudo-code. Introduction to C: Overview of C Program, Basic structure of a C Program. Constants, Variables & Data types: Character Set, Keywords & Identifiers. Control Statements, Functions, Structures and Unions.

UNIT-III **12 Hrs.**

Data Structure: Types of Data structures, Arrays, Queues, Linked list, Trees, Searching and Sorting Algorithm: Searching – Introduction, Linear search, Binary Search, Sorting - Introduction, bubble sort, Insertion sort, Selection sort, Merge sort. Comparisons of searching and sorting techniques.

UNIT-IV **12 Hrs.**

Database System concepts and architecture: Data Models, Schemas, and Instances, Three-schema architecture and Data Independence, Database Languages and Interfaces, The Database System Environment, Classification of Database Management Systems. Relational Data Model: Relational Model Concepts, Relational model Constraints and Relational Database Schemas, Update Operations, transactions and Dealing with Constraint Violations. SQL :Data manipulation in DBMS, Data types, SQL commands: Create Table, Inserting data, SELECT, DELETE, UPDATE, ALTER TABLE, DROP TABLE, RENAME, DESCRIBE.

REFERENCE BOOKS:

1. E. Balagurusamy, Programming in ANSI C, 7th Edition, Tata McGraw Hill.
2. Introduction to Information Technology ITL education solution Ltd, Second Edition
3. K.R. Venugopal and Sudeep R Prasad, Programming with C, 4th Edition, Tata McGraw-Hill Education.
4. M. Morris Mano, Digital Logic and Computer design, PHI, 2015
5. Thomas L Floyd, Digital Fundamentals, 10th Edition, Pearson, 2011.
6. Ramez Elmasri and Shamkanth B.Navate, Fundamentals of Database Systems, 7th Edition, Pearson Education 2.
7. Horowtz and Shani, Fundamentals of Data Structures in C, Universities Press, 2nd edition, 2008.

Note: *BRIDGE COURSE: Non - Credit course for only **Non - Computer Science Graduates**; [B.Sc. / B.A. / B.Com with Mathematics at 10 + 2 Level or Graduation level] (as per the norms of the concerned University [AICTE - NOTIFICATION]).

25MCAP101: Operating Systems Lab

Hours/Week: 6 I.A. Marks: 30
Credits: 3 Exam. Marks: 70

The practical problems shall be based on the theory concepts covered under operating systems course.

25MCAP102: Advanced Data Structures and Algorithms Lab

Hours/Week: 6 I.A. Marks: 30
Credits: 3 Exam. Marks: 70

The practical problems shall be based on the theory concepts covered under advanced data structures and algorithm course.

25MCAP103: .Net Technology Lab

Hours/Week: 6 I.A. Marks: 30
Credits: 3 Exam. Marks: 70

The practical problems shall be based on the theory concepts covered under .NET Technology course.

25MCAP104: Python Programming Lab

Hours/Week: 6 I.A. Marks: 30
Credits: 3 Exam. Marks: 70

The practical problems shall be based on the theory concepts covered under Python programming course.

25MCAH201: Principles of Data Science

Hours/Week: 4 I.A. Marks: 30
Credits: 4 Exam. Marks: 70

Course Learning Objectives:

- Introduce the theoretical foundation of data science.
- Understand data types, data preprocessing, and lifecycle management.
- Build a solid foundation in exploratory data analysis and statistical inference.
- Explore introductory machine learning algorithms and project workflows

Course Outcomes: After completing the course, the students will be able to,

- Understand the data science lifecycle and types of data.
- Describe data preprocessing techniques and challenges.
- Interpret and summarize data using statistical and visualization techniques.
- Apply statistical inference for data-driven decision making.
- Understand basic machine learning concepts and data science project frameworks.

UNIT-I **12 Hrs.**

Foundations of Data Science and Data Handling: Definition and Scope of Data Science; Data Science vs Data Analytics vs Data Engineering; Data Science Lifecycle: Collection, Cleaning, Exploration, Modeling, Interpretation; Types of Data: Structured, Semi-Structured, Unstructured; Overview of Data Sources: APIs, Open Datasets, Web Data; Data Preprocessing Concepts: Handling Missing Values, Duplicates, Outliers; Overview of Data Transformation Techniques: Normalization, Scaling, Encoding; Feature Engineering and Feature Selection.

UNIT-II **12 Hrs.**

Data Exploration and Visualization Principles: Role of Exploratory Data Analysis (EDA) in Data Science; Descriptive Statistics: Measures of Central Tendency and Dispersion; Concepts of Univariate, Bivariate, and Multivariate Analysis; Data Visualization - Histograms, Boxplots, Scatterplots, Heatmaps; Concepts of Correlation and Covariance; Ethical Considerations in Data Analysis (Bias, Fairness, Privacy).

UNIT-III **12 Hrs.**

Statistical Inference and Predictive Modeling Concepts - Basic Concepts of Probability and Random Variables; Probability Distributions: Normal, Binomial, Poisson (Theoretical Properties); Sampling Techniques, Population vs Sample; Estimation and Confidence Intervals; Hypothesis Testing: Concepts, Types of Errors, p-Value Interpretation; Linear Regression: Assumptions, Interpretation, Model Fit Measures; Logistic Regression: Use Cases and Conceptual Explanation; Bias-Variance Trade-off: Overfitting and Underfitting.

UNIT-IV **12 Hrs.**

Machine Learning and Data Science Project Framework: Machine Learning Paradigms: Supervised vs Unsupervised; Supervised Learning - K-Nearest Neighbors, Decision Trees, SVM; Unsupervised- K-Means clustering, Agglomerative clustering, Divisive clustering; Model Evaluation Metrics: Accuracy, Precision, Recall, F1-Score; Model Validation Techniques - Train/Test Split, Cross-Validation; Overview of Project Lifecycle - Problem Definition, Data Understanding, Modeling, Deployment Concepts; Introduction to Reproducibility, Version Control, Documentation in Data Science Projects.

REFERENCE BOOKS:

1. Jake VanderPlas, " Python Data Science Handbook"
2. Allen B. Downey, " Think Stats (2nd Edition)"
3. Gareth James et al. An Introduction to Statistical Learning
4. Allen B. Downey Elements of Data Science
5. Ethics and Data Science – https://dssg.uchicago.edu/wp-content/uploads/2019/05/Ethics-and-Data-Science.pdf
6. Kleinberg, Hopcroft, Kannan, Foundations of Data Science – https://www.cs.cornell.edu/jeh/book.pdf

25MCAH202: SOFTWARE ENGINEERING

Hours/Week: 4 I.A. Marks: 30
Credits: 4 Exam. Marks: 70

Course Learning Objectives:

- Be agile software developers with a comprehensive set of skills appropriate to the needs of the dynamic global computing-based society.
- Capable of team and organizational leadership in computing project settings, and have a broad understanding of ethical application of computing-based solutions to societal and organizational problems.
- Acquire skills and knowledge to advance their career, including continually upgrading professional, communication, analytic, and technical skills.
- To understand project scheduling concept and risk management associated to various type of projects.

Course Outcomes: After completing the course, the students will be able to,

- Recognize the software engineering and software process.
- Understand different activities of Software process.
- Realize the concepts of agile methods and software testing.
- Learn the techniques of functional and non-functional requirements.
- Familiar with concepts of detailed and object oriented design.
- Define various software application domains and remember different process model used in software development.
- An ability to apply engineering design to produce solutions that meet specified needs.
- Consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors

UNIT-I 12 Hrs.

Introduction: Professional Software Development, Software Engineering Ethics. Case Studies. Software Processes: Models. Process activities. Coping with Change. The Rational Unified Process.

UNIT-II 12 Hrs.

Agile Software Development: Agile methods. Plan-driven and agile development. Extreme programming. Agile project management. Scaling agile methods. Requirements Engineering: Functional and non-functional requirements. The software Requirements Document. Requirements Specification. Requirements Engineering Processes. Requirements Elicitation and Analysis. Requirements validation. Requirements Management, need for SRS, characteristics of SRS, organization of SRS document.

UNIT-III 12 Hrs.

Function Oriented Design: Design Principles, Module-Level Concepts, Design Notation and Specification, Structured Design Methodology, Verification, Metrics. Object-Oriented Design: OO Analysis and OO Design, OO Concepts, Design Concepts, Unified Modeling Language (UML), A Design Methodology, Metrics.

UNIT-IV **12 Hrs.**

Software Testing: Development Testing, Test-Driven Development, Release Testing, User Testing. Software Evolution: Evolution Processes. Program Evolution Dynamics. Software Maintenance. Legacy System Management. Project Planning: Software Pricing. Plan-Driven Development. Project Scheduling. Agile Planning. Estimation Techniques. Quality Management: Software Quality. Software Standards. Reviews and Inspections. Software Measurement and Metrics.

REFERENCE BOOKS:

1. Ian Sommerville, Software Engineering, 9th Edition, Pearson Education, 2012. (Listed topics from Chapters 1, 2, 3, 4, 5, 7, 8, 9, and 24)
2. Roger S. Pressman, Software Engineering-A Practitioners approach, 7th Edition, Tata McGraw Hill.
3. Pankaj Jalote, An Integrated Approach to Software Engineering, Wiley-India.

25MCAH203: DATA COMMUNICATIONS AND COMPUTER NETWORKS

Hours/Week: 4 I.A. Marks: 30
Credits: 4 Exam. Marks: 70

Course Learning Objectives:

- Acquire the computer networking knowledge as well as the existing connectivity technologies and the required infrastructure which comprises the key steps involved in the communication process.
- Identify the key issues for the realization of the LAN/WAN/MAN network architectures and the hybridized existing form in the business environment and enterprise.
- Establish a solid knowledge of the layered approach that makes design, implementation and operation of extensive networks possible. To learn the 7-layer OSI network model (each layer and its responsibilities) and understand the TCP/IP suite of protocols.
- Establish a solid knowledge of the layered approach that makes design, implementation, and operation of extensive networks possible.

Course Outcomes: After completing the course, the students will be able to,

- Understanding the basic communication concepts in real time applications
- Identify the different networking and internetworking devices and their functions within a network
- Familiar with the protocols in DC and CN and their future uses in various applications.
- Know the Importance of ISO - OSI and TCP / IP reference model and functions of each layer.
- Clearly understand the importance of services of all layers.
- Familiar with the architecture of a number of different networks and classifications.
- Gather the importance of all applications protocols and port specifications.

UNIT-I **12 Hrs.**

Introduction: Data Communications Fundamentals, Computer Communications Architecture, Data Communication tasks, Data Communication Systems Applications, Data Communication System Characteristic Features, Data Communication network criteria, Protocols and standards, Transmission mode, Analog and Digital Signals, Bit rate, Baud rate, Channel capacity using Nyquist and Shannon's relation. Modulation, encoding and decoding techniques. Transmission media characteristics, Transmission impairments, multiplexing.

UNIT-II **12 Hrs.**

Introduction to Computer Networks, Application and goals, Classification of Computer Networks, ISO-OSI Architecture, Services of Physical, Data link, Network, Transport, Session, Presentation and Application Layers., TCP /IP reference Model, Topology. Physical and Data Link Layer Services, Network Layer Services: Networking and Internetworking Technology Devices, Repeaters, Bridges, Routers, Gateways and Other Devices.

UNIT-III **12 Hrs.**

TCP/IP Protocol Suit: Overview of TCP/IP, TCP/IP and the Internet, TCP/IP and OSI, Internetwork Protocol (IP), Classes of IP, Addressing, Protocols in the Network Layer, Address Resolution Protocol (ARP), Reverse Address Resolution Protocol (RARP), Internet

Control MESSAGE Protocol (ICMP), Internet Group Message Protocol (IGMP), Transport Layer Services, Functionalities of the Transport Layer.

UNIT-IV **12 Hrs.**

Upper OSI Layers: Session Layer Services, SPDU. Presentation Layer Services: Application layer Services, PPDU. Application Layer Services: Client / Server Model,, BOOTP, Dynamic Host Configuration Protocol(DHCP), Domain Name System (DNS), Telnet, Fle transfer Protocol (FTP), Trivial File Transfer Protocol (TFTP),Simple Mail Transfer Protocol (SMTP), Post Office Protocol (POP), Simple Network Management Protocol (SNMP), Hyper Text Transfer Protocol (HTTP), World Wide Web (WWW).

REFERENCE BOOKS:

1. Prakash C. Gupta, Data Communications and Computer Networks, PHI (Latest Edition), 2013.
2. Behrouz A Forouzan, Data Communications and Networking, McGraw Hill, (Fourth Edition), 2007.
3. Behrouz A Forouzan and Firouz, Computer Networks A Top - Down Approach, McGraw Hill, (Special Indian Edition), 2012.
4. Tananbaum A.S., "Computer Networks", 3rd Ed, PHI, 1999.
5. Black U., "Computer Networks-Protocols, Standards and Interfaces", PHI, 1996.
6. Stallings W., "Computer Communication Networks", PHI.
7. Stallings W., "SNMP, SNMPv2, SNMPv3, RMON 1&2", 3rd Ed., Addison Wesley, 1999.

25MCAS201: ANDROID PROGRAMMING

Hours/Week: 4 I.A. Marks: 30
Credits: 4 Exam. Marks: 70

Course Learning Objectives:

- Fundamentals of Android Operating systems, android application components and android development framework.
- Designing of Android User Interfaces using various components like buttons, text views, toggle buttons, check boxes, spinners etc.
- How to develop software's with reasonable complexity and deploying software to mobile devices.
- The concept of intents and broadcasts, persistent storage and database connectivity concepts.

Course Outcomes: After completing the course, the students will be able to,

- Demonstrate their understanding of the fundamentals of Android operating systems
- Show their skills of using Android software development tools
- Develop software with reasonable complexity and their design aspects.
- Deploy software to mobile devices and debug the programs
- Understands the working of Android OS Practically and able to develop, deploy and maintain the Android Applications.
- Understands the concept of persistent storage and develop User Interface.
- Recognizes basics of SQLlite database and perform various possible operation on database.

UNIT-I **12 Hrs.**

Introduction to Android Operating System: Introduction to Mobile applications, Android OS design and Features – Android development framework, SDK features, Installing and running applications on Eclipse platform, Creating AVDs, Types of Android applications, Android tools Android application components – Android Manifest file, Externalizing resources like values, themes, layouts, Menus, Resources for different devices and languages, Runtime Configuration Changes Android Application Lifecycle – Activities, Activity lifecycle, activity states, monitoring state changes.

UNIT-II **12 Hrs.**

Android User Interface: Measurements – Device and pixel density independent measuring units Layouts – Linear, Relative, Grid and Table Layouts User Interface (UI) Components – Editable and non-editable Text Views, Buttons, Radio and Toggle Buttons, Checkboxes, Spinners, Dialog and pickers Event Handling – Handling clicks or changes of various UI components, Fragments – Creating fragments, Lifecycle of fragments, Fragment states, Adding fragments to Activity, adding, removing and replacing fragments with fragment transactions, interfacing between fragments and Activities, Multi-screen Activities.

UNIT-III **12 Hrs.**

Intents and Broadcasts: Intent – Using intents to launch Activities, Explicitly starting new, Activity, Implicit Intents, Passing data to Intents, Getting results from Activities, Native Actions, using Intent to dial a number or to send SMS Broadcast Receivers – Using Intent filters to service implicit Intents, Resolving Intent filters, finding and using Intents received within an Activity Notifications – Creating and Displaying notifications, Displaying Toasts.

UNIT-IV **12 Hrs.**

Android capabilities: flutter framework, Introduction-including libraries future, asynch and wait, including files in application shared preferences, Introduction to SQLLite database, creating and opening a database, creating tables, inserting, retrieving and deleting data, Registering Content Providers, Using content Providers (insert, delete, retrieve and update). Connecting to internet resource, using download manager Location Based Services – Finding Current Location and showing location on the Map, updating location.

REFERENCE BOOKS:

1. RetoMeier,,Wiley India, (Wrox) , Professional Android 4 Application Development, 2012.
2. James C Sheusi, Android Application Development for Java Programmers, Cengage Learning, 2013.
3. Wei-MengLee, Beginning Android 4 Application Development ,Wiley India (Wrox), 2013.
4. Rap Payne, Beginning App Development with Flutter: Create Cross-Platform Mobile Apps -

25MCAS202: WIRELESS SENSOR NETWORKS

Hours/Week: 4 I.A. Marks: 30
Credits: 4 Exam. Marks: 70

Course Learning Objectives:

- To understand the basic WSN technology and supporting protocols, with emphasis placed on standardization basic sensor systems and provide a survey of sensor technology.
- Understand the medium access control protocols and address physical layer concerns.
- Learn key routing protocols for sensor networks and main design issues.
- Understand the Sensor management, sensor network middleware, operating systems.

Course Outcomes: After completing the course, the students will be able to,

- Learn Ad hoc network and Sensor Network fundamentals.
- Understand the different routing protocols and the uses.
- Have an in-depth knowledge on sensor network architecture and design issues.
- Understand the transport layer and security issues possible in Ad hoc and Sensor networks.
- Have an exposure to mote programming platforms and tools.
- To develop wireless sensor systems for different applications using.
- Demonstrate knowledge of routing protocols developed for WSN.

UNIT-I **12 Hrs.**

AD – HOC Networks, Introduction and Routing Protocols, Elements of Ad hoc Wireless Networks, Issues in Ad hoc wireless networks, Example commercial applications of Ad hoc networking, Ad hoc wireless Internet, Issues in Designing a Routing Protocol for Ad Hoc Wireless Networks, Classifications of Routing Protocols, Table Driven Routing Protocols – Destination Sequenced Distance Vector (DSDV), On–Demand Routing protocols –Ad hoc On–Demand Distance Vector Routing (AODV).

UNIT-II **12 Hrs.**

Sensor Networks, Introduction and Architecture, Challenges for Wireless Sensor Networks, Enabling Technologies for Wireless Sensor Networks, WSN application examples, Single-Node Architecture – Hardware Components, Energy Consumption of Sensor Nodes, Network Architecture – Sensor Network Scenarios, Transceiver Design Considerations, Optimization Goals and Figures of Merit.

UNIT-III **12 Hrs.**

WSN Networking concepts and protocols, MAC Protocols for Wireless Sensor Networks, Low Duty Cycle Protocols And Wakeup Concepts – S-MAC, The Mediation Device Protocol, Contention based protocols – PAMAS, Schedule based protocols – LEACH, IEEE 802.15.4 MAC protocol, Routing Protocols Energy Efficient Routing, Challenges and Issues in Transport layer protocol.

UNIT-IV **12 Hrs.**

Sensor Network Security, Network Security Requirements, Issues and Challenges in Security Provisioning, Network Security Attacks, Layer wise attacks in wireless sensor networks, possible solutions for jamming, tampering, black hole attack, flooding attack. Key Distribution and Management, Secure Routing – SPINS, reliability requirements in

sensor networks. Sensor network platforms and tools. Sensor Node Hardware – Berkeley Motes, Programming Challenges, Node-level software platforms – TinyOS, nesC, CONTIKIOS, Node-level Simulators – NS2 and its extension to sensor networks, COOJA, TOSSIM, Programming beyond individual nodes – State centric programming.

REFERENCE BOOKS:

1. C. Siva Ram Murthy and B. S. Manoj, Ad Hoc Wireless Networks Architectures and Protocolsǁ, Prentice Hall, PTR, 2004. (UNIT I).
2. HolgerKarl, Andreas willig, Protocol and Architecture for Wireless Sensor Networksǁ, John wiley publication, Jan 2006.(UNIT II-V).
3. Feng Zhao, Leonidas Guibas, Wireless Sensor Networks: an information processing approach, Elsevier publication, 2004.
4. Charles E. Perkins, Ad Hoc Networkingǁ, Addison Wesley, 2000.
5. I.F. Akyildiz, W. Su, Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, computer networks, Elsevier, 2002.

25MCAE201: DISTRIBUTED COMPUTING

Hours/Week: 3 I.A. Marks: 30
Credits: 3 Exam. Marks: 70

Course learning objectives: Students will able to know about

- Abstraction and details of file systems.
- Contemporary knowledge in parallel and distributed computing.
- Performance and flexibility issues related to systems design decisions.
- Methodologies and approaches for reasoning about concurrent and distributed programs.

Course Outcomes: After completing the course, the students will be able to,

- Understand the foundations of distributed systems
- Study issues related to clock Synchronization and the need for global state
- Acquire distributed mutual exclusion and deadlock detection algorithms.
- Understand the significance of agreement, fault tolerance and recovery protocols.
- Learn the characteristics of peer-to-peer and distributed shared memory systems.
- Come across the real information about distributed mutual exclusion strategies.
- Realize the importance of check points and usages.

UNIT-I **9 Hrs.**

Introduction: Definition –Relation to computer system components –Motivation –Relation to parallel systems – Message-passing systems versus shared memory systems –Primitives for distributed communication –Synchronous versus asynchronous executions –Design issues and challenges. A model of distributed computations: A distributed program –A model of distributed executions –Models of communication networks –Global state – Cuts –Past and future cones of an event –Models of process communications. Logical Time: A framework for a system of logical clocks –Scalar time –Vector time – Physical clock synchronization: NTP.

UNIT-II **9 Hrs.**

Message ordering and group communication: Message ordering paradigms – Asynchronous execution with synchronous communication –Synchronous program order on an asynchronous system –Group communication – Causal order (CO) - Total order. Global state and snapshot recording algorithms: Introduction –System model and definitions –Snapshot algorithms for FIFO channels.

UNIT-III **9 Hrs.**

Distributed mutual exclusion algorithms: Introduction – Preliminaries – Lamport's algorithm – Ricart-Agrawala algorithm – Maekawa's algorithm – Suzuki–Kasami's broadcast algorithm. Deadlock detection in distributed systems: Introduction – System model – Preliminaries – Models of deadlocks – Knapp's classification – Algorithms for the single resource model, the AND model and the OR model.

UNIT-IV **9 Hrs.**

Check pointing and rollback recovery: Introduction – Background and definitions – Issues in failure recovery – Checkpoint-based recovery – Log-based rollback recovery – Coordinated check-pointing algorithm – Algorithm for asynchronous check-pointing and recovery. Consensus and agreement algorithms: Problem definition – Overview of results – Agreement in a failure – free system – Agreement in synchronous systems with failures.

Peer-to-peer computing and overlay graphs: Introduction – Data indexing and overlays – Chord – Content addressable networks – Tapestry

REFERENCE BOOKS:

1. Kshemkalyani, Ajay D., and Mukesh Singhal. Distributed computing: principles, algorithms, and systems. Cambridge University Press, 2011.
2. George Coulouris, Jean Dollimore and Tim Kindberg, Distributed Systems Concepts and Design, Fifth Edition, Pearson Education, 2012.
3. Pradeep K Sinha, "Distributed Operating Systems: Concepts and Design", Prentice Hall of India, 2007.
4. Mukesh Singhal and Niranjan G. Shivaratri. Advanced concepts in operating systems. McGraw-Hill, Inc., 1994.
5. Tanenbaum A.S., Van Steen M., Distributed Systems: Principles and Paradigmsll, Pearson Education, 2007.

25MCAE202: ADVANCED JAVA PROGRAMMING

Hours/Week: 3 I.A. Marks: 30
Credits: 3 Exam. Marks: 70

Course Learning Objectives:

- Explore the fundamentals of exception handling in Java.
- Create and use exception handling through classes and objects.
- Understand the concept of Threads in Java.
- To learn event handling in Swings, JFrames and Components.

Course Outcomes: After completing the course, the students will be able to,

- Use the Java language for writing well-organized, complex computer programs with both command line and graphical user interfaces.
- Develop web application using Java Servlet and Java Server Pages technology.
- Learn how to work with ODBC, JSP and Servlets.
- Develop sophisticated, interactive user interfaces using the Java Swing class and appropriate layout managers.
- Understand advanced topics including multithreading, internet networking.
- Gathering the JDBC database connectivity, Java beans importance and services.
- Come across the applications of java events and their approaches.

UNIT-I **9 Hrs.**

Review on Basics of Java Technology, Exception and Multithreads: Exception-type, Uncaught Exception, Using try-catch, throw, throws, finally, Throwable class and object, Exception classes, Create own exception subclass. Creating multiple threads, isAlive(), join(), Thread priorities, synchronization, - Deadlock, wait(), notify(), notify All() methods, Inter-Thread Communication, suspend, resume & stop the threads. Swing: Introduction to Swing, Event Handling, Component Organizers: JApplet, Handling Swing Controls like Icons JFrames, Lists , Tables, Trees, Text Components, Progress Indicators.

UNIT-II **9 Hrs.**

JDBC: Presentation to JDBC CONNECTION settings – The Concept of JDBC – JDBC Driver Types – JDBC Packages – A Brief Overview of the JDBC Process – Database Connection – Associating the JDBC/ODBC Bridge with the Database – Statement Objects – Result Set, metadata, Transaction. JSP: Introduction, disadvantages, JSP v/s Servlets, Lifecycle of JSP, Comments, JSP documents, JSP elements, Action elements, implicit objects, scope, character quoting conventions, unified expression language.

UNIT-III **9 Hrs.**

Enterprise Java Bean: Preparing a Class to be a JavaBean, Creating a JavaBean, JavaBean Properties, Types of beans, Stateful Session bean, Stateless Session bean, Entity bean. Servlet API and Lifecycle: Background, The Life Cycle of a Servlet & The JSDK-A Simple Servlet – The Servlet API -RolePlay-Servlet Concept – The javax.servlet Package – Reading Servlet Parameters, The javax.servlet.http Package – Handling HTTP Request and Responses – Using Cookies – Session Tracking.

UNIT-IV **9 Hrs.**

HIBERNATE: Introduction, Writing the application, application development approach, creating database and tables in MySQL, creating a web application, Adding the required library files, creating a java bean class, creating hibernate configuration and mapping file, adding a mapping resource, creating JSPs. WEB Services: SOAP, Building a web services using JAX-WS, Building web service. JAVAMAIL: Mail Protocols, Components of the Javamail API, JAVAMAIL API, Starting with API.

REFERENCE BOOKS:

1. Naughton and H.Schildt, Java 2-The complete reference Fifth Edition McGraw Hill, (2007).
2. Sharanam Shah, Vaishali Shah, Java EE 6 for Beginners, SPD
3. Herbert Schildt, Java Complete Reference, Seventh Edition, TMH. (Unit I)
4. Shah, Java EE Project using EJB 3, JPA and struts 2 for beginners, SPD
5. C Xavier, Java Programming A practical Approach, McGraw Hill

25MCAE203: OBJECT ORIENTED DATA MODELING

Hours/Week: 3 I.A. Marks: 30
Credits: 3 Exam. Marks: 70

Course learning objectives:

- To incorporate the idea of object oriented concepts for solving system modeling and design problems.
- Design and implement object oriented models using UML appropriate notations.
- The convert class into and object and apply the object oriented methodologies to design cleaner software from the problem statement.
- To calculate the performance of the system by applying the suitable and reusable system design.

Course Outcomes: After completing the course, the students will be able to,

- Demonstrate the ability to apply the knowledge of object oriented concepts for solving system modeling and design problems.
- Design and implement object oriented models using UML appropriate notations.
- Ability to apply the concepts of object oriented methodologies to design cleaner software from the problem statement.
- Apply the concept of domain and application analysis for designing UML Diagrams.
- Comprehend the concept of architectural design approaches for system design and implementation issues for object oriented models.
- Uderstand the meaning of a pattern and different Pattern categories– Relationships between patterns.
- Come across the importance of class design specifications and use cases.

UNIT-I **9 Hrs.**

Importance of modelling, Principles of modelling– Object Oriented Modelling– Object Orientation– Object Oriented Development and Themes - OO methodology– Three Models OO Themes– Abstraction– Encapsulation– Combining data and behavior– modelling as Design techniques - Brief overview of OMT by Rumbaugh– Introducing the UML – overview– conceptual model of the UML– Architecture– Software Development Life Cycle. Class Modelling and Advanced Class Modelling: Object and class concepts.

UNIT-II **9 Hrs.**

System Conception: Devising a system concept–Elaborating a concept– Preparing a problem statement– Domain Analysis: Overview of analysis– Domain class model– Domain state model– Domain interaction model–Iterating the analysis– Application Analysis: Application interaction model– Application class model– Application state model–adding operations.

UNIT-III **9 Hrs.**

System Design: Overview of system design–Estimating performance–Making a reuse plan–Breaking a system in to sub-systems– Identifying concurrency– Allocation of sub-systems– Management of data storage–Handling global resources–Choosing a software control strategy– Handling boundary conditions– Setting the trade-off priorities–Common architectural styles–Architecture of the ATM system as the example.

UNIT-IV **9 Hrs.**

Class Design: Overview of class design– Bridging the gap– Realizing use cases–Designing algorithms– recursing downwards–Refactoring– Design optimization–Reification of behaviour– Adjustment of inheritance–Organizing a class design–ATM example– Implementation Modelling: Overview of implementation– Fine-tuning classes– Fine-tuning generalizations– Realizing associations– Testing– Design patterns: What is a pattern and what makes a pattern? Pattern categories– Relationships between patterns –Pattern descriptions. Communication Patterns: Forwarder-Receiver; Client-Dispatcher-Server; Publisher-Subscriber.

REFERENCE BOOKS:

1. James Rumbaugh et.al, Object-Oriented Modelling and Design, PHI, 1991.
2. Grady Booch et.al, Object-Oriented Analysis and Design with Applications, 2007, Wesley, 3rd Edition.
3. Michael. Blaha, James, Rumbaugh, "Object-Oriented Modeling and Design with UML", 2nd Edition, Pearson Education, 2005.
4. Mark. Priestley, "Practical Object-Oriented Design with UML", 2nd Edition, Tata McGraw-Hill, 2003.

25MCAE204: CLOUD COMPUTING

Hours/Week: 3 I.A. Marks: 30
Credits: 3 Exam. Marks: 70

Course Learning Objectives:

- Characteristics and design principles of grid and cloud computing.
- Security mechanisms in grid and cloud computing applications.
- Designing methodologies of distributed computing and Importance of cloud computing environments.
- The concepts of virtualization and use of cloud service models.

Course Outcomes: After completing the course, the students will be able to,

- Demonstrate in-depth understanding characteristics of grid and cloud computing.
- Demonstrate an in-depth understand of the design principles of grid and cloud computing.
- Illustrate security mechanisms in grid and cloud computing applications.
- Design and demonstrate distributed computing applications.
- Understand the importance of cloud computing environments.
- Understand cloud based data storage, cloud based database solutions and research trends in cloud computing.
- Analyze cloud security issues and applications of Fog computing.

UNIT-I **9 Hrs.**

Cloud computing basics: - Cloud computing components- Infrastructure-services-storage applications database services – Deployment models of Cloud- Services offered by Cloud- Benefits and Limitations of Cloud Computing – Issues in Cloud security- Cloud security services and design principles.

UNIT-II **9 Hrs.**

Virtualization fundamentals: Virtualization – Enabling technology for cloud computing- Types of Virtualization- Server Virtualization- Desktop Virtualization – Memory Virtualization – Application and Storage Virtualization- Tools and Products available for Virtualization.

UNIT-III **9 Hrs.**

SAAS and PAAS: Getting started with SaaS - Understanding the multitenant nature of SaaS solutions- Understanding OpenSaaS Solutions- Understanding Service Oriented Architecture- PaaS- Benefits and Limitations of PaaS. Security as a Service.

UNIT-IV **9 Hrs.**

IAAS and cloud data storage: - Understanding IaaS- Improving performance through Load balancing- Server Types within IaaS solutions- Utilizing cloud based NAS devices – Understanding Cloud based data storage- Cloud based database solutions- Cloud based block storage. Cloud Applications and security: Open Source and Commercial Clouds, Cloud Simulators, Research trends in Cloud Computing, Fog Computing and applications, Cloud Security challenges.

REFERENCE BOOKS:

1. R. Buyya, C. Vecchiola, S T. Selvi, Mastering Cloud Computing, McGraw Hill (India) Pvt Ltd., 2013
2. Kris Jamsa, Cloud Computing: SaaS, PaaS, IaaS, "Virtualization, Business Models, Mobile, Security and more, Jones & Bartlett Learning Company, 2013
3. Ronald L.Krutz, Russell vines, Cloud Security: A Comprehensive Guide to Secure Cloud Computing, Wiley Publishing Inc., 2010.
4. Gautam Shroff, Enterprise Cloud Computing - Technology, Architecture, Applications, Cambridge University Press, 2010
5. Anthony T .Velte, Toby J.Velte, Robert Elsenpeter, Cloud Computing: A Practical Approach, Tata McGraw Hill Edition, Fourth Reprint, 2010
6. Ronald L. Krutz, Russell Dean Vines, Cloud Security: A Comprehensive Guide to Secure Cloud Computing, Wiley- India, 2010.
7. Antonopoulos, Nick; Gillam, Lee, Cloud Computing Principles, Systems and Applications, Springer, 2010.
8. G. Reese, Cloud Application Architecture, O'Reilly, 2009.

25MCAP201: Principles of Data Science Lab

Hours/Week: 6	I.A. Marks: 30
Credits: 3	Exam. Marks: 70

The practical problems shall be based on the theory concepts covered under principles of data science course.

25MCAP202: Data Communications and Computer Networks Lab

Hours/Week: 6	I.A. Marks: 30
Credits: 3	Exam. Marks: 70

The practical problems shall be based on the theory concepts covered under Data communications and computer networks course.

25MCAP203: Android Programming Lab

Hours/Week: 6	I.A. Marks: 30
Credits: 3	Exam. Marks: 70

The practical problems shall be based on the theory concepts covered under Android programming course.

25MCAP204: Wireless Sensor Networks Lab

Hours/Week: 6	I.A. Marks: 30
Credits: 3	Exam. Marks: 70

The practical problems shall be based on the theory concepts covered under Wireless Sensor Networks course.

25MCAP205: Advanced Java Programming Lab

Hours/Week: 6	I.A. Marks: 30
Credits: 3	Exam. Marks: 70

The practical problems shall be based on the theory concepts covered under Advanced Java Programming course.

25MCAH301: MACHINE LEARNING

Hours/Week: 4 I.A. Marks: 30
Credits: 4 Exam. Marks: 70

Course Learning Objectives:

- Understand theoretical foundations of machine learning.
- Master implementation of classical and modern ML algorithms.
- Design and evaluate ML models for real-world problems.
- Explore practical applications through lab sessions and mini-projects.

Course Outcomes: After completing the course, the students will be able to,

- Recognize concept of knowledge representation and predicate logic and transform the real life information in different representation.
- Realize the state space and its searching strategies.
- Understand machine learning concepts and range of problems that can be handled by machine learning.
- Apply the machine learning concepts in real life problems.
- Compare AI with human intelligence and traditional information processing and discuss its strengths and limitations as well as its application to complex and human-centred problems.
- Discuss the core concepts and algorithms of advanced AI, including informed searching Algorithm, Different Types of Machine Learning Approaches
- Apply the basic principles, models, and algorithms of AI to recognize, model, and solve problems in the analysis and design of information systems.

UNIT-I **12 Hrs.**

Introduction to Machine Learning and Deep Learning, Life Cycle, Learning paradigms: supervised, unsupervised, reinforcement, Basic concepts: hypothesis space, generalization, overfitting, Case study discussion: ML/DL applications in real-world.

UNIT-II **12 Hrs.**

Perceptron, Multi-Layer Perceptrons (MLPs), Activation functions: Sigmoid, ReLU, Tanh, GELU, Loss functions, Optimization Algorithms: Adam, Adagrad, RMSProp; Backpropagation, Gradient Descent: Stochastic and Minibatch Gradient; Preventing Overfitting in Neural Networks.

UNIT-III **12 Hrs.**

Deep Neural Networks – Filters and Feature Maps – Description of Convolutional Layer, Maxpooling – Convolution Network Architecture – Image Classification; Deep Architectures: LeNet, AlexNet, VGG, ResNet, DenseNet, Transfer learning and fine-tuning.

UNIT-IV **12 Hrs.**

Recurrent Neural Networks, Bidirectional RNNs; Modern RNNs: Gated Recurrent Units (GRU), Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (BLSTM), Deep Recurrent Neural Network, Generative Adversarial Networks; GAN variants; Autoencoder: Architecture, Denoising and Sparcity.

REFERENCE BOOKS:

1. George F Luger, Artificial Intelligence – Structures and Strategies for Complex problem solving, 5thEdn, pearson.
2. E. Rich, K. Knight, S B Nair, Artificial intelligence, 3rdEdn, McGraw Hill.
3. S. Russel and P. Norvig, Artificial intelligence – A Modern Approach, 3rdEdn, Pearson
4. D W Patterson, Introduction to Artificial Intelligence and Expert Systems, PHI, 1990.
5. EthemAlpaydin, Introduction to Machine Learning- 3rd Edition, PHI.
6. Tom M. Mitchell, Machine Learning, McGraw-Hill.
7. Ian Goodfellow and YoshuaBengio and Aaron Courville, Deep Learning (Adaptive Computation and Machine Learning), MIT Press, 2016.

25MCAH302: INTERNET OF THINGS

Hours/Week: 4 I.A. Marks: 30
Credits: 4 Exam. Marks: 70

Course Learning Objectives:

- Understand the concepts of Internet of Things.
- Analyze basic protocols in wireless sensor network.
- Design IoT applications in different domain and be able to analyze their performance.
- Implement basic IoT applications on embedded platform.

Course Outcomes: After completing the course, the students will be able to,

- Understand IoT applications and Architectures in real world and realize the various IoT Protocols
- Differentiate between the levels of the IoT stack and be familiar with the key technologies
- Interface different sensors to arduinouno and raspberry pi to read the environment data.
- Appreciate the role of big data, cloud computing and data analytics in a typical IoT system
- To provide an overview on the ICT ecosystem and enabling environment
- To provide an understanding of the technologies and the standards relating to the IoT
- To develop skills on IoT technical planning and Identify how IoT differs from traditional data.

UNIT-I **12 Hrs.**

Introduction to Internet of Things –Definition and Characteristics of IoT, Physical Design of IoT – IoT Protocols, IoT communication models, IoT Communication APIs IoT enabled Technologies – Wireless Sensor Networks, Cloud Computing, Big data analytics, Communication protocols, Embedded Systems, IoT Levels and Templates Domain Specific IoTs – Home, City, Environment, Energy, Retail, Logistics, Agriculture, Industry, health and Lifestyle, IoT challenges.

UNIT-II **12 Hrs.**

IoT and M2M – Software defined networks, network function virtualization, difference between SDN and NFV for IoT, SNMP, Basics of IoT System Management with NETCOZF, YANG- NETCONF, YANG, NETOPEER.

UNIT-III **12 Hrs.**

Sensors and Actuators : Types of Sensors-Temperature and Humidity (DHT11), Proximity (Ultrasonic, IR), Motion (PIR, Accelerometer) and Environmental (Gas). Actuators: Relays, Servo Motors, DC Motors. Role of actuators in IoT Automation.

Programming with Raspberry PI: Introduction to Raspberry Pi, Interfacing sensors and actuators with Python programs. **Introduction to Arduino microcontroller**: Components of Arduino uno, Interfacing Sensors and actuators with Arduino Uno. **Introduction to Nodemcu**: architecture of ESP8266, programming with Nodemcu.

Edge Computing in IoT: Definition and Need for Edge Computing, Edge vs. Fog vs. Cloud Computing, Benefits of Edge Computing in IoT.

UNIT-IV **12 Hrs.**

IoT Physical Servers and Cloud Offerings – Introduction to Cloud Storage models and communication APIs Webserver – Web server for IoT, Cloud for IoT, Python web application framework, Designing a RESTful web API.

REFERENCE BOOKS:

1. Arshdeep Bahga and Vijay Madisetti, Internet of Things - A Hands-on Approach,Universities Press, 2015, ISBN: 9788173719547
2. Matt Richardson & Shawn Wallace, Getting Started with Raspberry Pi, O'Reilly (SPD),2014, ISBN: 9789350249759.

25MCAH303: ADVANCED DATABASE MANAGEMENT SYSTEMS

Hours/Week: 4 I.A. Marks: 30
Credits: 4 Exam. Marks: 70

Course Learning Objectives:

- Basics of NoSQL databases, Relational Databases, Information Retrieval and XML databases.
- The concepts of column databases, distributed database and data warehousing schemes
- Various concepts of MongoDB and types of consistency.
- Advance Databases, Convergent databases and Disruptive Databases.

Course Outcomes: After completing the course, the students will be able to,

- Explore the concepts of NoSQL Databases.
- Understand and use columnar and distributed database patterns.
- Learn to use various Data models for a variety of databases.
- Explore the relationship between Big Data and NoSQL databases
- Work with NoSQL databases to analyze the big data for useful business applications.
- Understands the concept of MongoDB and types of consistency.
- Learn the concepts of Advance Databases, Convergent databases and Disruptive Databases.

UNIT-I **12 Hrs.**

Database Revolutions- System Architecture- Relational Database- Database Design, Data Storage- Transaction Management- Data warehouse and Data Mining- Information Retrieval, Big Data evolution- CAP Theorem- Birth of NoSQL, Document Database, XML and XML Databases- JSON Document Databases- Graph Databases.

UNIT-II **12 Hrs.**

Column Databases, Data Warehousing Schemes- Columnar Alternative- Sybase IQ- CStore and Vertica - Column Database Architectures, SSD and In-Memory Databases, In-Memory, Databases- Berkeley Analytics Data Stack and Spark.

UNIT-III **12 Hrs.**

Distributed Database Patterns, Distributed Relational Databases- Non-relational Distributed Databases- MongoDB - Sharing and Replication- HBase- Cassandra- Consistency Models, Types of Consistency- Consistency MongoDB- HBase Consistency- Cassandra Consistency.

UNIT-IV **12 Hrs.**

Data Models and Storage- SQL- NoSQL APIs- Return SQL - Advance Databases PostgreSQL- Riak- CouchDB- NEO4J- Redis- Future Databases— Revolution Revisited- Counter revolutionaries- Oracle HQ- Other Convergent Databases- Disruptive Database Technologies.

REFERENCE BOOKS:

1. Abraham Silberschatz, Henry F. Korth, S. Sudarshan, "Database System Concepts", Sixth Edition, McGrawHill.
2. Guy Harrison, "Next Generation Databases",Apress, 2015.
3. Eric Redmond, Jim R Wilson, "Seven Databases in Seven Weeks", LLC. 2018.
4. Dan Sullivan, "NoSQL for Mere Mortals", Addison-Wesley, 2015.
5. Adam Fowler, "NoSQL for Dummies ", John Wiley & Sons, 2015.

25MCAS301: DIGITAL IMAGE PROCESSING

Hours/Week: 4 I.A. Marks: 30
Credits: 4 Exam. Marks: 70

Course Learning Objectives:

- Understand digital image representation and processing fundamentals.
- Apply techniques for image enhancement, restoration, segmentation, and compression.
- Implement state-of-the-art and deep learning-based algorithms.
- Solve real-world image processing problems using modern libraries.

Course Outcomes: After completing the course, the students will be able to,

- Understand the need for image transforms different types of image transforms and their properties.
- Develop any image processing application and understand the rapid advances in Machine vision.
- Learn different techniques employed for the enhancement of images.
- Identify different causes for image degradation and overview of image restoration techniques.
- Explain different Image enhancement techniques
- Design & Synthesize Color image processing and its real world applications.
- Come across the image representation with their model approaches.

UNIT- I **12 Hrs.**

Digitized image and its properties: Basic concepts, Image digitization, Digital image properties. Image Preprocessing: Image pre-processing; Histogram processing, Enhancement using arithmetic / logic operations, Basics of spatial filtering, Smoothing spatial filters, Sharpening spatial filters. Brightness and geometric transformations, local preprocessing, Mathematical tools: matrix operations, 2D signals, convolution

UNIT-II **12 Hrs.**

Edge Detection Techniques: Roberts, Sobel, Canny, Laplacian of Gaussian, Line Detection algorithms; Segmentation methods: Thresholding, Edge-based segmentation, Region based segmentation, Matching.

UNIT-III **12 Hrs.**

Image Enhancement: Image enhancement in the frequency domain: Background, Introduction to the Fourier transform and the frequency domain, Smoothing Frequency-Domain filters, Sharpening Frequency Domain filters, Homomorphic filtering. Image Compression: Fundamentals of compression: redundancy, entropy, rate-distortion, Lossless compression: Run Length Encoding, Huffman coding, Arithmetic coding, LZW, Lossy compression: Transform coding, JPEG, JPEG2000.

UNIT-IV **12 Hrs.**

Image Representation and Description: Region identification, Contour-based shape representation and description, Region based shape representation and description, Shape classes. Feature descriptors: SIFT, SURF, ORB, Texture analysis: GLCM, LBP, Image moments, HOG descriptors. Mathematical Morphology: Basic morphological concepts, Morphology principles, Binary dilation and erosion, Gray-scale dilation and erosion, Morphological segmentation and watersheds.

REFERENCE BOOKS:

1. Milan Sonka, Vaclav Hlavac and Roger Boyle, Image Processing, Analysis and Machine Vision 2nd Edition, Thomson Learning, 2001.
2. Rafel C Gonzalez and Richard E Woods, Digital Image Processing, 2nd Edition, Pearson Education, 2003.
3. Anil K Jain, Fundamentals of Digital Image Processing Pearson Education/Prentice-Hall of India Pvt. Ltd., 1997.
4. B. Chanda, D Dutta Majumder, Digital Image Processing and Analysis Prentice-Hall India, 2002.

25MCAS302: PRINCIPLES OF CYBER SECURITY

Hours/Week: 4 I.A. Marks: 30
Credits: 4 Exam. Marks: 70

Course Learning Objectives:

- Fundamental concepts of Information systems and Information security.
- Analyze the basic algorithms used for image processing &image compression with morphological image processing.
- To study the image fundamentals and mathematical transforms necessary for image processing.
- Design algorithms to solve image processing problems and meet design specifications.

Course Outcomes: After completing the course, the students will be able to,

- Identify information security goals, classical encryption techniques and acquire fundamental knowledge on the concepts of finite fields and number theory.
- Understand, compare and apply different encryption and decryption techniques to solve problems related to confidentiality and authentication
- Apply the knowledge of cryptographic checksums and evaluate the performance of different message digest algorithms for verifying the integrity of varying message sizes
- Apply different digital signature algorithms to achieve authentication and create secure applications
- Apply network security basics, analyze different attacks on networks and evaluate the performance of firewalls and security protocols like SSL, IPSec, and PGP

UNIT-I **12 Hrs.**

Introduction to Cyber Security - Cyberspace and Cyber security, Tools and Methods used in Cybercrime: Proxy Server and Anonymizers, Password Cracking, Key loggers and Spyware, Virus and Worms, Trojan and backdoors, Steganography, DOS and DDOS attack, SQL injection, Buffer Overflow, Attacks on mobile device/cell phones. Standards of Good Practice for Information Security, ISO Suite of Information Security Standards, NIST Cyber Security Framework and Security Documents.

UNIT-II **12 Hrs.**

Information Risk Assessment: Asset Identification, Threat Identification, Vulnerability Identification, Risk Assessment Approaches. System Access: User Authentication, Password-Based Authentication, Possession-Based Authentication, Biometric Authentication, Risk Assessment for User Authentication, Access Control, Customer Access.

UNIT-III **12 Hrs.**

Phishing and Identity Theft Introduction - Methods of Phishing, Phishing Techniques, Phishing Toolkits and Spy Phishing. Identity Theft PII, Types of Identity Theft, Techniques of ID Theft. Network Defence tools: Firewalls and Packet Filters: Firewall Basics, Packet Filter Vs Firewall, How a Firewall Protects a Network, Packet Characteristic to Filter, Stateless Vs Stateful Firewalls, Network Address Translation (NAT) and Port Forwarding, The Basics of Virtual Private Networks.

UNIT-IV **12 Hrs.**

Threat and Incident Management: Technical Vulnerability Management, Security Event Logging, Security Event Management, Threat Intelligence, Cyber Attack Protection. Digital Forensics Science, Need for Computer Cyber forensics and Digital Evidence, Digital Forensics Life Cycle. The Legal perspective: The Indian IT Act, Challenges to Indian Law and cybercriminal scenario in India, Amendments to the Indian IT Act.

REFERENCE BOOKS:

1. William Stallings, Effective Cyber Security: A Guide to Using Best Practices and Standards,Addison-Wesley Professional, ISBN-13: 978-0134772806.
2. Mike Shema, Anti-Hacker Tool Kit (Indian Edition), 4th Edition, Publication McGraw Hill, ISBN:9789339212155.
3. Nina Godbole and SunitBelpure, Cyber Security Understanding Cyber Crimes, Computer Forensicsand Legal Perspectives, Wiley Publication, ISBN 9788126521791.

25MCAE301: MOBILE COMPUTING

Hours/Week: 3 I.A. Marks: 30
Credits: 3 Exam. Marks: 70

Course Learning Objectives:

- The computer systems perspective on the converging areas of wireless networking, embedded systems, and software
- To provide an overview of Wireless Communication networks area and its applications in communication engineering.
- The contribution of Wireless Communication networks to overall technological growth.
- Explain the various terminology, principles, devices, schemes, concepts, algorithms and different methodologies used in Wireless Communication Networks.

Course Outcomes: After completing the course, the students will be able to,

- Discuss cellular radio concepts and identify various propagation effects.
- Have knowledge of the mobile system specifications.
- Classify multiple access techniques in mobile communication.
- Outline cellular mobile communication standards and analyze various methodologies to improve the cellular capacity.
- Explain the principles and theories of mobile computing technologies and describe infrastructures and technologies of mobile computing technologies.
- List applications in different domains that mobile computing offers to the public, employees, and businesses.
- Describe the possible future of mobile computing technologies and applications.

UNIT-I **9 Hrs.**

Introduction to Mobile Computing: Applications, A Simplified Reference Model, Wireless Transmission: Frequencies of Radio Transmission, Signals, Antennas, Signal Propagation, Multiplexing, Modulation, Spread Spectrum, Cellular System. Media Access Control: Motivation for a Specialized MAC, SDMA, FDMA, TDMA, CDMA, and Comparisons.

UNIT-II **9 Hrs.**

Telecommunications Systems: GSM-Mobile Services, System Architecture, Radio Interface, Protocol, Security, DECT- System Architecture, Protocol Architecture, Wireless LAN: Infrared V/S Radio Transmission, Infrastructure And Ad-Hoc Networks, IEEE 802.11, HPERLAN, Bluetooth.

UNIT-III **9 Hrs.**

Mobile Network Layer: Mobile IP, Dynamic host configuration protocol, Mobile ad-hoc networks- Routing, Destination sequence distance vector, Dynamic source routing.Mobile Transport Layer: Traditional TCP, classical TCP improvements, TCP over 2.5/3G wireless networks.

UNIT-IV **9 Hrs.**

Support For Mobility: File Systems, World Wide Web, Wireless Application Protocol (WAP)-Architecture, Wireless Datagram Protocol, Transport Layer Security, Wireless Transaction Protocol, Wireless Session Protocol, Wireless Application Environment, Wireless Markup Language (WML), WML Script and WAP 2.0.

REFERENCE BOOKS:

1. Jochen Schiller, Mobile Communications, PHI, Second Edition, 2003.
2. Prasant Kumar Pattnaik, Rajib Mall, Fundamentals of Mobile Computing, PHI Learning Pvt.Ltd, New Delhi, 2012.
3. Dharma Prakash Agarval, Qing and An Zeng, "Introduction to Wireless and Mobile systems", Thomson Asia Pvt Ltd, 2005.
4. Uwe Hansmann, Lothar Merk, Martin S. Nicklons and Thomas Stober, Principles of Mobile Computing, Springer, 2003.
5. William.C.Y.Lee, Mobile Cellular Telecommunications, Analog and Digital Systems, Second Edition, TataMcGraw Hill Edition, 2006.
6. C.K.Toh, AdHoc Mobile Wireless NetworksII, First edition, Pearson Education, 2002.

25MCAE302: SOFTWARE QUALITY ASSURANCE

Hours/Week: 3 I.A. Marks: 30
Credits: 3 Exam. Marks: 70

Course Learning Objectives:

- To study fundamental concepts in software testing, including software testing objectives, process, criteria, strategies, and methods.
- To discuss various software testing issues and solutions in software unit test; integration, regression, and system testing.
- To learn how to planning a test project, design test cases and data, conduct testing operations, manage software problems and defects, generate a testing report.
- To expose the advanced software testing topics, such as object-oriented software testing methods, and component-based software testing issues, challenges, and solutions.

Course Outcomes: After completing the course, the students will be able to,

- Identify and apply various software metrics, which determines the quality level of software
- Identify and evaluate the quality level of internal and external attributes of the software product
- Compare and Pick out the right reliability model for evaluating the software
- Evaluate the reliability of any given software product
- Design new metrics and reliability models for evaluating the quality level of the software based on the requirement
- Applying software testing knowledge and methods to practice-oriented software testing projects.
- Understand software test automation problems and solutions.

UNIT-I **9 Hrs.**

Software Quality: Basics, Popular Views, Quality Professional Views, Software Quality, Total Quality Management and Summary. Fundamentals of Measurement Theory: Definition, Operational Definition and Measurement, Level of Measurement, Some Basic Measures, Reliability and Validity, Measurement Errors, Criteria for Causality, Summary. Software Quality Metrics Overview: Product Quality Metrics, In-Process Quality Metrics, Metrics for Software Maintenance, Examples for Metrics Programs, Collecting Software Engineering Data.

UNIT-II **9 Hrs.**

Applying The Seven Basic Quality Tools In Software Development: Ishikawa's Seven Basic Tools, Checklist, Pareo Diagram, Histogram, Run Charts, Scatter Diagram, Control Chart, Cause And Effect Diagram. The Rayleigh Model: Reliability Models, The Rayleigh Model Basic Assumptions, Implementation, Reliability And Predictive Validity.

UNIT-III **9 Hrs.**

Complexity Metrics and Models: Lines of Code, Halstead's Software Science, Cyclomatic Complexity Syntactic Metrics, An Example of Module Design Metrics in Practice .Metric And Lessons Learned for Object Oriented Projects: Object Oriented Concepts and Constructs, Design And Complexity Metrics, Productivity Metrics, Quality And Quality Management Metrics, Lessons Learned For object oriented Projects. Availability Metrics: Definition and Measurement of System Availability, Reliability Availability and Defect

Rate, Collecting Customer Outage Data for Quality Improvement, In-Process Metrics for Outage and Availability.

UNIT-IV **9 Hrs.**

Conducting Software Project Assessment: Audit Ad Assessment, Software Process Maturity Assessment and Software Project Assessment, Software Process Assessment Software Process Improvement: Measuring Process Maturity, Measuring Process Capability, Staged Versus Continuous Debating Religion, Measuring the Value of Process Improvement, Measuring Process Compliance, Using Function Point Metrics to Measure Software Process Improvement: Software Process Improvement Sequences, Process Improvement Economies, Measuring Process Improvement at Activity Levels

REFERENCE BOOKS:

1. Stephen H Khan, Metrics and Models in Software Quality Engineering, Pearson 2nd edition 2013.
2. Norman E-Fentor and Share Lawrence Pflieger, Software Metrics, International Thomson, Computer Press 1997.
3. S.A. Kelkar, Software Quality and Testing Market,. PHI Learing, Pvt, Ltd 2012.

25MCAE303: BLOCK CHAIN MANAGEMENT

Hours/Week: 3 I.A. Marks: 30
Credits: 3 Exam. Marks: 70

Course Learning Objectives:

- Basics of block chain management and Fundamentals of the design principles of Bitcoin and Ethereum.
- Advantages of Block chain over distributed computing.
- Solutions of soft computing algorithms for optimization.
- Designing, building and deploying smart contracts and distributed applications.

Course Outcomes: After completing the course, the students will be able to,

- Understand the fundamentals of the design principles of Bitcoin and Ethereum.
- Explain the Simplified Payment Verification protocol.
- Interact with a block chain system by sending and reading transactions.
- Evaluate the solutions of soft computing algorithms for optimization.
- Design build and deploy smart contracts and distributed applications.
- Easily Analyze regulations of crypto currency.
- Evaluate roots of bitcoin and the applications of crypto currency.

UNIT-I **9 Hrs.**

Basics of Block Chain Management, Distributed Database, Two General Problem, Byzantine General Problem and Fault Tolerance, Hadoop Distributed File System, Distributed Hash Table, ASIC resistance, Turing Complete, Cryptography: Hash function, Digital Signature - ECDSA, Memory Hard Algorithm, Zero Knowledge Proof.

UNIT-II **9 Hrs.**

Blockchain: Introduction, Advantage over Conventional Distributed Database, Blockchain Network, Mining Mechanism, Distributed Consensus, Merkle Patricia Tree, Gas Limit, Transactions and Fee, Anonymity, Reward, Chain Policy, Life of Blockchain application, Soft & Hard Fork, Private and Public Blockchain.

UNIT-III **9 Hrs.**

Distributed Consensus: Nakamoto consensus, Proof of Work, Proof of Stake, Proof of Burn, Difficulty Level, Sybil Attack, Energy utilization and alternate. Crypto currency: History, Distributed Ledger, Bitcoin protocols - Mining strategy and rewards, Ethereum - Construction, DAO, Smart Contract, GHOST, Vulnerability, Attacks, Sidechain, Namecoin.

UNIT-IV **9 Hrs.**

Crypto Currency Regulations: Stakeholders, Roots of Bit Coin, Legal Aspects-Crypto Currency Exchange, Black Market and Global Economy. Applications: Internet of Things, Medical Record Management System, Domain Name Service and Future of Blockchain.

REFERENCE BOOKS:

1. Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller and Steven Goldfeder, Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction, Princeton University Press, 2016.
2. Antonopoulos, Mastering .Bitcoin: Unlocking Digital Cryptocurrencies
3. Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System
4. Dr. Gavin Wood, "ETHEREUM: A Secure Decentralized Transaction Ledger," Yellow paper.2014.
5. Nicola Atzei, Massimo Bartoletti, and TizianaCimoli, A survey of attacks on Ethereum smart contracts.

25MCAE304: NATURAL LANGUAGE PROCESSING

Hours/Week: 3 I.A. Marks: 30
Credits: 3 Exam. Marks: 70

Course Learning Objectives:

- Understand the foundational concepts and linguistic challenges involved in NLP, especially for multilingual and Indian languages.
- Apply standard text preprocessing techniques and tokenization schemes to prepare text for NLP tasks.
- Explain and compare classical text representation techniques and embeddings, including contextual and non-contextual word representations.
- Implement key NLP tasks using classical models (e.g., POS tagging, NER, text classification, topic modeling).
- Analyze the structure and functionality of neural architectures such as RNNs, LSTMs, and Transformers.
- Use and fine-tune modern large language models (e.g., BERT, GPT) for downstream applications like summarization and question answering.
- Critically evaluate the performance and limitations of traditional and modern NLP techniques.

Course Outcomes: After completing the course, the students will be able to,

- Describe the fundamentals of NLP and the linguistic challenges specific to Indian and multilingual contexts.
- Perform text preprocessing and tokenization using classical and modern tokenization schemes.
- Construct and compare word, sentence, and document embeddings using BoW, TF-IDF, Word2Vec, GloVe, and BERT.
- Build classical NLP models for tasks such as language modeling, classification, POS tagging, and NER.
- Explain and analyze neural architectures like RNNs, LSTMs, and Transformers used in modern NLP.
- Utilize pretrained LLMs (BERT, GPT, etc.) for real-world applications and perform basic prompt engineering.
- Evaluate NLP models based on performance metrics and interpret their limitations in different domains.

UNIT-I **12 Hrs.**

Foundations of NLP: Introduction to NLP and its applications, Challenges of processing Indian languages; Challenges in NLP – Ambiguity, Data Sparsity, Domain Adaptation; Text Preprocessing - Normalization, Lemmatization, Stemming, Stopword Removal; Tokenization – Types of Tokenization - Character, Word, ngrams, Subword, and Sentence Tokenization; Tokenization schemes **-** Whitespace tokenization, punctuation-based tokenization, Byte Pair Encoding (BPE), WordPiece.

UNIT-II **12 Hrs.**

Text Representation Techniques: Basic Representations - Bag-of-Words (BoW), TF, TF-IDF; Limitations, One-hot Encoding vs. Dense Representations; Word Embeddings - Word2Vec, GloVe, FastText; Limitations; Contextual Embeddings - ELMo, BERT, GPT; Sentence and Document Embeddings - SBERT, USE.

UNIT-III **12 Hrs.**

NLP Tasks and Classical Models: Language Modeling - n-gram models, Laplace Smoothing; Text Similarity and Clustering; Topic Modeling – LDA; Text Classification - Sentiment Analysis, Spam Detection; Part-of-Speech (POS) Tagging, Named Entity Recognition (NER).

UNIT-IV **12 Hrs**.

Neural Architectures and Large Language Models: Sequence Models: RNN, LSTM, GRU – Concepts and Limitations, Transformer Architecture - Self-Attention, Positional Encoding, BERT and its variants (RoBERTa, ALBERT, DistilBERT); GPT Models (GPT-2, GPT-3, GPT-4) - Capabilities and Limitations; Prompt Engineering and Fine-tuning Pre-trained Models; Applications of LLMs - Summarization, Translation, Q&A.

REFERENCE BOOKS:

1. Daniel Jurafsky and James H. Martin. 2020. Speech and Language Processing. 3rd Edition (draft)
2. Christopher D. Manning and Hinrich Schütze. 1999. Foundations of Statistical Natural Language Processing. MIT Press.
3. Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta, Harshit Surana. 2020. Practical Natural Language Processing. O'Reilly.
4. Hobson Lane, Cole Howard, Hannes Hapke. 2019. Natural Language Processing in Action. Live Book.
5. Denis Rothman - Transformers for NLP
6. Yoav Goldberg - Neural Network Methods for NLP
7. Natural Language Processing with Python – Bird, Klein, Loper
8. Hugging Face Docs: https://huggingface.co/learn

25MCAP301: MACHINE LEARNING LAB

Hours/Week: 6 I.A. Marks: 30
Credits: 3 Exam. Marks: 70

The practical problems shall be based on the theory concepts covered under Artificial Intelligence & Machine Learning course.

25MCAP302: Internet of Things Lab

Hours/Week: 6 I.A. Marks: 30
Credits: 3 Exam. Marks: 70

The practical problems shall be based on the theory concepts covered under Internet of Things course.

25MCAP303: ADBMS LAB

Hours/Week: 6 I.A. Marks: 30
Credits: 3 Exam. Marks: 70

The practical problems shall be based on the theory concepts covered under ADBMS course.

25MCAP304: Digital Image Processing Lab

Hours/Week: 6 I.A. Marks: 30
Credits: 3 Exam. Marks: 70

The practical problems shall be based on the theory concepts covered under Image Processing course.

25MCAP305: Natural Language Processing Lab

Hours/Week: 6 I.A. Marks: 30
Credits: 3 Exam. Marks: 70

The practical problems shall be based on the theory concepts covered under Natural Language Processing course.

25MCAM301: Mini Project and Domain Knowledge Seminar

Hours/Week: 6 I.A. Marks: 30
Credits: 3 Exam. Marks: 70

The students shall take up the mini-project work based on their areas of advanced computing domains and in consultation with the mentor to carry out mini project works. A seminar shall be given on the advanced technologies and developments in computer science field.

25MCAP401: Dissertation and Viva-Voce

Hours/Week: 32 I.A. Marks: 100
Credits: 16 Max. Marks: 400

Dissertation & Viva Examination:

Dissertation Valuation – 200 marks
Viva - Voce - 100 marks
Total: 300 marks

Course Learning Objectives:

- To offer students a glimpse into real world problems and challenges that is needed in IT industries
- To enable students to create very precise specifications of the IT solution to be designed.
- To introduce students to the vast array of literature available of the various research/project challenges in the field of IT.
- To create awareness among the students of the characteristics of several domain areas where IT can be effectively used.

Course Outcomes: After completing the course, the students will be able to,

- Discover potential research areas in the field of IT.
- Conduct a survey of several available literature in the preferred field of study.
- Compare and contrast the several existing solutions for research/project challenge.
- Demonstrate an ability to work in teams and manage the conduct of the research/project study.
- Formulate and propose a plan for creating a solution for the research/project plan identified.
- Report and present the findings of the study conducted in the preferred domain.
- Improve communication and management skills of the students.