Jyoti Nivas College Autonomous

Post Graduate Centre
Hosur Road, Bangalore -95

Phone No: 255012143 Fax: 25533886
e-mail:jncpgcourses@gmail.com
Website: www.jyotinivas.org

Proposed Syllabus

Master of Computer Applications (MCA) – 2 year Programme

w. e. f. 2022-24 Batch onwards

CONTENTS

MCA PROGRAMME

The Master of Computer Applications (MCA) is a Post Graduate Programme started in the year 2003, affiliated to Bangalore City University, and approved by All India Council of Technical Education (AICTE).

PROGRAMME OBJECTIVES

The MCA programme aims to provide quality education in computer science and applications with the objective of equipping the students with the necessary skills that makes them competent to face the challenges and demands of rapidly expanding IT industry.

The curriculum ensures holistic development of the student, moulds them into capable leaders of tomorrow and socially committed citizens, developing their independent thinking and confidence, while imparting technical skills required for an IT profession.

The curriculum is designed to be flexible and wide-ranging, incorporating cutting edge technologies while ensuring a firm grasp of core fundamentals.

PROGRAMME OUTCOMES

In the span of four semesters, the students are well equipped with programming skills, analytical ability, research skills and strong knowledge in the domain. The students will be able to:

- Apply the knowledge of computer application to find solutions for real-life applications
- Analyse, design, develop and maintain software applications for different verticals
- Utilize the knowledge of computing technology with commitment on social, ethical, and cyber values
- Communicate her ideas and solutions effectively and to work in collaborative manner with others in a team
- Acquaint with the contemporary trends in industrial/research settings and thereby innovate novel solutions to existing problems
- Make use of modern tools and technologies to provide optimized solutions
- Recognize the need for continuing professional development and engage in life-long learning.

PROGRAMME STRUCTURE

- The MCA Course has a curriculum with syllabi consisting of (i) Core courses, (Theory & Practical) (ii) Skill Enhancement Courses, (iii) Discipline specific Elective Courses (iv) Open Electives and (v) Project work / Internship.
- In addition, add-on courses, group activities, soft skills, communication skills and value added training are provided based on the need.
- The total number of credits assigned to the course is 98 and the credits per paper are distributed as follows:
    - 4 credits per core theory courses with 4 hours of lecture/tutorial per week
    - 2 credits per practical courses with 4 hours of practical/project per week
    - 4 credits for discipline specific elective courses with 4 hours of lecture/tutorial per week
    - 2 credits per skill enhancement courses with 4 practical hours per week
    - 4 credits per open elective paper with 4 lecture hours per week
    - 12 credits for M.C.A. Internship Project

- Core courses are compulsory subjects offered by the department. Total credits for Core Courses including theory, practical papers, skill enhancement courses and projects are 82.
- Elective courses may be chosen by the student from the list of Electives offered by the department. Total credits for Elective Courses are 12.
- Open Elective has to be taken from other post graduate disciplines. Credit for open Elective course is 4.

REGULATION AND SCHEME

1. Eligibility: Candidate should have passed BCA/Bachelor Degree in Computer Science/ Engineering or equivalent Degree.

OR

Candidate should have passed B.Sc. /B.Com. /B.A. with Mathematics at 10+2 Level or at Graduation Level (with additional bridge courses as per the norms of the concerned University). Candidate should also have obtained at least 50% marks (45% marks in case of candidates belonging to reserved category) in the qualifying Examination.

2. Duration of the Course: 2 years (**4** semesters)

3. Medium of Instruction: English

4. Intake: 45

5. Mode of Admission:

Candidate must appear for Post Graduate Common Entrance Test **– PGCET** as per the schedule notified by Karnataka Examination Authority

OR

Candidate must appear for Karnataka Management Aptitude Test **– KMAT** conducted by Karnataka Post Graduate Private Colleges‟ Association.

END SEMESTER EXAMINATION

1. End Semester Examination will be conducted for the total marks of 100 and the time duration is 3 hours.
2. The Syllabus is divided in to five major units; the questions will be equally distributed among all the units.

Question Paper Pattern

Part A
A student has to answer 10 questions out of 12. Each question carries 4 marks

Part B
A student has to answer 6 questions out of 10. Each question carries 10 marks.

CONTINOUS INTERNAL ASSESMENT (CIA) PATTERN

CIA for Theory

Component	**Marks**
Mid Term Examination	15
Assignment / Presentation / Case Study / Mini Project	10
Group Activity	5
Total	**30**

CIA for Practical

Component	**Marks**
Mid Term Examination	15
Review of the work done in the class	10
Documentation	5
Total	**30**

GRADUATE ATTRIBUTES

1. Domain proficiency

Generally defined, domain proficiency implies knowledge and understanding of the essential aspects of a specific field of inquiry. Domain proficiency is used to evaluate the inputs, guide the process, and evaluate the end products within the context of value and validity.

2. Applied knowledge

It's about the description and demonstration of procedures and tasks that are learnt in the classroom. The knowledge could be applied to develop or revise procedures or algorithms to address a problem / situation.

3. Joint Effort

Joint effort is the collaborative effort of a group to achieve a common goal or to complete a task in the most effective and efficient way. This concept is seen within the greater framework of a team, which is a group of interdependent individuals who work together towards a common goal.

4. Problem solving

Problem solving is the act of defining a problem; determining the cause of the problem; identifying, prioritizing, and selecting alternatives for a solution; and implementing a solution.

5. Design and development

The design and development process should examine all the potential challenges and hurdles the students will need to overcome to develop an effective application. It involves defining the specification and the design that is simple yet creative and user friendly.

6. Research skills

Research skills refer to the ability to search for, locate, extract, organise, evaluate and use or present information that is relevant to a particular topic. It involves intensive search, investigation, and critical analysis, usually in response to a specific research question or hypothesis.

7. Application Governance.

Application Governance refers to the branch of project management dedicated to the planning, scheduling, resource allocation, execution, tracking and delivery of software and web projects.

8. Usage of modern tools.

Introducing the latest and open source tools that are available to handle the application development.

9. Innovation and Entrepreneurship: Identify a timely opportunity and using innovation to pursue that opportunity to create value and wealth for the betterment of the individual and society at large.

SEMESTER-WISE SCHEDULE

Semester	Course Category	Course Code	Course Title	Hours / Week (32)			Credits	Marks		
				L	T	P		CIA	ESE	Total
I	DSC	22MCA101	Operating System with Linux	3	1	–	4	30	70	100
	DSC	22MCA102	Data Structures and Algorithms	4	–	–	4	30	70	100
	DSC	22MCA103	Mathematical Foundations for Computer Applictions	4	–	–	4	30	70	100
	DSC	22MCA104	Database Management System	4	–	4	6	30+15	70+35	150
	DSC	22MCA105	Object Oriented Concepts using Java	4	–	4	6	30+15	70+35	150
	SEC	22MCA106	Problem Solving Techniques	–	–	4	2	15	35	50
			TOTAL	**19**	**1**	**12**	**26**	**195**	**455**	**650**
II	DSC	22MCA201	Software Engineering Methodologies	3	1	–	4	30	70	100
	DSC	22MCA202	Computer Networks	4	–	–	4	30	70	100
	DSC	22MCA203A/B/C	Elective I	3	1	–	4	30	70	100
	DSC	22MCA204	Web Technologies	4	–	4	6	30+15	70+35	150
	DSC	22MCA205	Mobile Application Development	4	–	4	6	30+15	70+35	150
	SEC	22MCA206	Exploring Python	–	–	4	2	15	35	50
			TOTAL	**18**	**2**	**12**	**26**	**195**	**455**	**650**
III	DSC	22MCA301	Cloud Computing	3	1	–	4	30	70	100
	DSE	22MCA302A/B/C	Elective II	3	1	–	4	30	70	100
	DSC	22MCA303	Data Analytics	4	–	4	6	30+15	70+35	150
	DSC	22MCA304	Research Methodology	4	–	4	6	30+15	70+35	150
	SEC	22MCA305	Mini Project	–	–	4	2	15	35	50
	OE	22MCA306	Open Elective	4	–	–	4	30	70	100
			TOTAL	**18**	**2**	**12**	**26**	**195**	**455**	**650**
IV	DSE	22MCA401	Advanced DBMS Concepts	4	–	–	4	30	70	100
	DSE	22MCA402A/B/C	Elective III	4	–	–	4	30	70	100
		22MCA403	Internship Project	–	–	–	12	100	200	300
			TOTAL	**8**	**0**	**0**	**20**	**160**	**340**	**500**

Total Credits (1- 4 semesters): 98

List of Discipline Specific Electives

Elective I	
22MCA203A	Artificial Intelligence
22MCA203B	Internet of Things (IoT)
22MCA203C	Digital Image Processing
Elective II	
22MCA302A	Machine Learning Techniques
22MCA302B	Cyber Security
22MCA302C	UI/UX Design
Elective III	
22MCA402A	Natural Language Processing
22MCA402B	Blockchain Technology
22MCA402C	Software Testing

List of Skill Enhancement Courses

- Problem Solving Techniques
- Exploring Python
- Agile Software Engineering

List of Open Elective Courses available for MCA Students

- Personal Wealth Management
- Approaches to Texts
- Life Style Management
- Chemistry in Daily Life

Open Elective offered to other Discipline Students

- Cyber Security

List of Mandatory/Non-Credit Courses

- Bridge Course: Foundations of Computer Applications (Semester I)
    - Applicable only to students from non-computer science background
- MOOC Courses (Semester II)
- Placement Training (Semester III)
- Research Paper presentation / Publication (Semester IV)

Add-on Certificate Courses

- Logic N Life
- Soft Skills & Communication Skills
- Academic Writing
- Recent Technologies

Semester I

22MCA101 OPERATING SYSTEM WITH LINUX

Total No. of Hours: 60 **Hours per week: L:T:P (3:1:0)**

Course Objectives: Operating system is essential part of any computer system. This paper focuses on the core structure, functions and design principles of operating system. The course provides comprehensive understanding of the layered architecture of LINUX operating system and its file system. It also focuses on acquiring skills needed to make effective use of Linux utilities, writing shell scripts, process and signal management and inter process communication.

Learning Outcomes:
After completion of this course, the student will be able to:

- Get the knowledge of process
- To implement Scheduling algorithms
- Handle Deadlock issues
- Make use of Linux utilities
- Design shell programs for different applications
- Design signal generation and handling programs

UNIT I: Operating system structures: services, user operating system interface, system calls, types of system calls, system programs, operating system design and implementation, operating system structure, virtual machines, OS debugging, OS generations and system boot.
Process Management: Processes, Process concept, Process scheduling, Operations on processes, Inter process communication **(12)**

UNIT II: CPU Scheduling: Basic concepts –Scheduling criteria - Scheduling algorithms - Multiple processor scheduling - Real time scheduling - Algorithm evaluation.
Process synchronization: background, the critical section problem, Peterson's solution, semaphores, classic problems of synchronization
Deadlocks: System model - Deadlock characterization - Methods for handling deadlocks - Deadlock prevention - Deadlock avoidance - Deadlock detection - Recovery from deadlock**.** **(12)**

UNIT III: Memory management: Background – Swapping - Contiguous memory allocation – Paging – Segmentation - Segmentation with paging. Virtual memory: Background - Demand paging - Process creation - Page replacement -Allocation of frames - Thrashing.
Storage Management: directory and disk structure, file system mounting, file sharing, protection.
Mass storage structure: overview, disk structure, disk attachment, disk scheduling, disk management, swap space management, RAID structure. **(12)**

UNIT IV: Linux Utilities-File handling utilities, Security by file permissions, Process utilities, , Filters-grep, more, less, wc, cmp, diff, nl, head, tail, sort, comm, advanced filters sed – scripts, operation, addresses, commands, applications, awk – execution, fields and records, scripts, operation, patterns, actions, functions, using system commands in awk.
Working with shell(bash): Introduction, shell responsibilities, pipes and input Redirection, output redirection, here documents, running a shell script, the shell as a programming language, shell meta characters, file name substitution, shell variables, command substitution, shell commands, the environment, quoting, test command, control structures, arithmetic in shell, shell script examples, interrupt processing, functions, debugging shell scripts.

(12)

UNIT V : Process – Process concept, Kernel support for process, process attributes, process control - process creation, waiting for a process, process termination, zombie process, orphan process, Process APIs. Signals– Introduction to signals, Signal generation and handling, Kernel support for signals, Signal function, unreliable signals, reliable signals, kill, raise , alarm, pause, abort, sleep functions.

(12)

Reference Books:

1. Silberschatz, Galvin, "Operating System Concepts", WSE, 9th Edition, 2010.
2. Andrew S. Tanenbaum, "Modern Operating Systems", 3rd edition, 2008.
3. William Stallings, "Operating Systems", 4th Edition, Pearson Education, 2001.
4. N.Matthew, R.Stones, "Beginning Linux Programming", 4th Edition, ,Wrox, Wiley India Edition.
5. Robert Love, "Linux System Programming", O'Reilly, SPD.
6. Sumitabha Das, "Unix Concepts and Applications", 4th Edition, TMH.

22MCA102 DATA STRUCTURES AND ALGORITHMS

Total No. of Hours: 60 **Hours per week: L:T:P (4:0:0)**

Course Objectives: The paper focuses on the abstract characterizations as well as the design and implementation of data structures such as arrays, stacks, queues, linked lists, binary search trees, and graphs along with algorithms. Algorithms will be analysed for their asymptotic behavior in terms of time complexity and space requirements will be considered as well.

Learning Outcomes:
After completion of this course, students will be able to:

- Have improving understanding of iterative and recursive methods.
- Understand a core group of basic data structures as enumerated in topics below.
- Be able to conceptualize many programming issues at a higher level through data structures.
- Know the tradeoffs of each studied data structure so as to employ the appropriate one for a given situation.
- Be able to incorporate data structures for efficient handling of data into their programs.
- Be able to code algorithms involving data structures using a C programming language.
- Be able to analyze data structures and their algorithms for asymptotic behavior.

UNIT 1: Introduction - Abstract Data Types (ADTs), Specification and Implementation, Classification of Data Structures: Primitive and Non- Primitive, Linear and Nonlinear; Algorithms - Asymptotic Analysis and Notation: "Big-Oh", "Big-Theta", "Big-Omega", Iterative and Recursive Algorithms: Towers of Hanoi; Computation Time Complexity and Space Complexity analysis; Divide and Conquer Technique: Introduction - Sorting - Quick Sort, Merge Sort; Strassen's Matrix Multiplication. **(12)**

UNIT II: Linked List: Definition, Representation, Types: Singly Linked List. Linked list as a data Structure, Inserting and removing nodes from a list, Doubly linked List, Circular Linked List.
Stack: Definition, Representation, Operations and Applications: Evaluation of Arithmetic Expressions, Evaluation of postfix expression, infix to prefix, postfix to infix conversion. Array Implementations: Fixed size and resizable. Linked List Implementation. **(12)**

UNIT III: Queue: The Queue ADT, Array Implementations, Circular Array Implementation, Linked List Implementation.
Binary Search Trees: Definitions for Binary Tree and Binary Search Tree, Properties of Binary Trees, Full, Balanced, Complete Binary Trees, Implementing Binary Trees using Linked nodes, Binary Tree Traversal Techniques. **(12)**

UNIT IV: Graph ADT and Definitions - Data Structures and Implementation issues for Graphs, Graph Traversal: Breadth First and Depth First Traversal, Greedy Algorithms: Introduction, Minimum Spanning Trees, Shortest Path: Dijkstra's Algorithm **(12)**

UNIT V: Dynamic Programming, All pairs shortest path problem, Traveling salesman problem. Optimal- parameterisation for product of sequence of matrices. Introduction to P, NP, NP complete problems, NP hard problems

Approximation Algorithm – Vertex cover; Parallel Algorithms – PRAM, Pointer jumping – List ranking. **(12)**

Reference Books:

1. Robert L. Kruse, Bruce P. Leung, Clovis L. Tondo, "Data structures and program design in C", BPB Publications, 2010.
2. Mark Allen Weiss, "Data structures and Algorithm Analysis in C", Tata McGraw hill Publishing, 2nd edition, 2006.
3. E. Horowitz and Sahani, "Fundamentals of computer algorithms", Universities Press, 2nd Edition, 2008.
4. Sara Baase and Allen Van Gelder, "Computer algorithms, Introduction to Design & Analysis", Pearson Education, 3rd edition, 2000.
5. SartajSahni, Data structures, "Algorithms and Applications in C++", University Press, 2nd edition, 2005.

22MCA103 MATHEMATICAL FOUNDATIONS FOR COMPUTER APPLICATIONS

Total No. of Hours: 60 **Hours per week: L:T:P (4:0:0)**

Course Objectives: The aim of this course is to provide the grounding knowledge of various mathematical concepts and its application in computer science domain.

Course Outcome

- Understand the concept and applications of vector spaces, subspaces and linear independence.
- Understand various inner products and able to perform various inner product operations.
- Explore the applicability of general Linear Transformations, Linear operators, Composition of operators and linear transformations.
- Understand the basics of probability theory and its applications

Unit-I: Set Theory and Matrices

Sets, Operations on sets, Cardinality of sets, inclusion-exclusion principle, pigeonhole principle, matrices, finding Eigen values and Eigen vectors. **(12)**

Unit-II: Vector Spaces: General Vector Spaces, Subspaces, Linear Independence, Basis and Dimension, Span, Fundamental Theorems, Row Space, Column Space, Nullspace, Rank and Nullity, Four Fundamental Spaces. Inner Product Spaces: General Inner Products, Euclidean and Weighted Inner Product, Length, Distance, Norm, Angle and Orthogonality in Inner Product Spaces, Caucy-Schwarz Inequality, Orthogonal Complement, Orthonormal Bases, Gram-Schmidt Procedure. **(12)**

Unit-III: Linear Transformations: General Liner Transformations, Linear operators, Composition of operators and linear transformations, Kernel and Range, Dimension theorem for Linear Transformation, Inverse Linear Transformations, Matrices of General Linear Transformations. **(12)**

Unit-IV: **Probability Theory -** Basic definitions, additive law (with proof), independence of events conditional probability, multiplicative law through definition of conditional probability, Bayes' theorem and its applications. **(12)**

UNIT V: Graphs: Introduction, Graphs and Graph models, Graph Terminology and Special types of Graphs , Representing Graphs & Graph Isomorphism, Connectivity, Euler and Hamilton Paths, Shortest Path Problems, Planar Graphs, Graph Coloring -Chromatic Polynomials.

Trees: Introduction, Properties-Rooted Trees- Spanning Trees, Minimum Spanning Trees. **(12)**

Reference Books:

1. Howard Anton and Chris Rorres, "Elementary Linear Algebra", John Wiley and Sons, 9th Edition, 2008.
2. Douglas C. Montgomery and George C. Runger, "Applied Statistics and Probability for Engineers", John Wiley and Sons, 3rd Edition, 2003.
3. Ronald E Walpole, Raymond H Myers, Sharon L Myers, “Probability & Statistics for engineers and scientists”, 6th Edition, Prentice Hall.
4. Trivedi et.al, “Probability and statistics with computer applications”, Tata McGraw Hill.
5. Gupta S.C & Kapoor V.K, “Fundamentals of Mathematical statistics”, Sultan Chand & sons, 2009.
6. “Kenneth H.Rosen, “Discrete Mathematics and Its Applications”, Tata McGraw-Hill, 5th Edition 2003.

22MCA104 DATABASE MANAGEMENT SYSTEM [60 Hrs]

Total No. of Hours: 60 + 60 Hours per week: L:T:P(4:0:4)

Course Objectives: This course sets a strong foundation for the design and development of data base applications. It helps the students to acquire the basics of databases, different data models and familiarize with the relational and object-oriented database technology.

Learning Outcomes:
On completion of this course, the student will be able to:

- Perform data modelling and work with popular RDBMS databases such as MySQL, SQL server
- Work with DDL and DML commands
- Implement Stored Procedures, Triggers, Functions, Cursors
- Compare and contrast various database systems

UNIT I: Purpose of database systems, view of data - data abstraction, instances and schemas, data models, Database languages -DDL & DML, Database architecture – two tier & three tier, Data mining and information retrieval. **(12)**

UNIT II: Storage structure and file organization: Primary and secondary storage devices, sequential, indexed sequential, random file access, hashing techniques.
E-R data model: Entities, attributes and relationships, different types of attributes, Drawing E-R diagrams. **(12)**

UNIT III: Relational Data model: Relation, Integrity constraints-domain, entity and referential integrity constraints, Relational algebra, select, project and join operations, Normalization concepts, first, second, third normal forms, Boyce-Codd normal form, Projection-Join normal form.

(12)

UNIT IV: SQL: Data Definition & Data Manipulation Commands, Sub queries, Correlated sub-query, Concept of a view, functions, stored procedures, and triggers.

Data Administration: Introduction, security issues, different methods of protecting the data base, database encryption, types of database failure, recovery and concurrency control, Locking Mechanism. **(12)**

UNIT V: Object Oriented Database Development: Introduction, defining a class, attribute, user structures, operations, range for an attribute, relationships, defining an abstract class, creating object instances, object query language. **(12)**

Suggested List of Lab Exercises:

- Introduction to Database, Database Architecture, Data models
- Database languages - DDL & DML
- E-R data model, Drawing E-R diagrams
- Relational data model - constraints, relational algebra
- Normalization concepts
- SQL: Data Definition Commands - Creation of database, tables with the constraints, drop, truncate, alter commands
- Data Manipulation Commands, Data Query language and all related commands, group by, order by, having clauses
- Aggregate functions, Sub queries and Joins
- Concept of a view – Creation of views, operations
- Introduction to Stored Procedures, Triggers, Functions, and Cursors.

Reference Books:

1. Elmasri & Navathe, Addison Wesley, "Fundamentals of database systems", Pearson Education, 7th Edition, 2016.
2. Fred R McFadden, Jeffrey A Hoffer and Mary B Prescott, "Modern database management system", 8th edition, 2010.
3. C J Date, "An Introduction to database systems", Pearson Education, 8th Edition, 2009.
4. Abraham Silberschatz, Henry F.Korth, S.Sudarshan, "Database System Concepts", Tata McGraw Hill Publication, 7th Edition, 2019.

22MCA105 OBJECT ORIENTED CONCEPTS USING JAVA

Total No. of Hours: 60 + 60 **Hours per week: L:T:P(4:0:4)**

Course Objectives: The objective of this paper is to teach the fundamentals of the Java language. Java is a pure object-oriented language, language of the Internet.

Learning Outcomes:

After completion of this course, the student will be able to:

- Design problem solutions using Object Oriented Techniques
- Apply the concepts of data abstraction, encapsulation and inheritance for problem solutions
- Write and execute Java programs based on collection framework and hibernate

UNIT I: Introduction to object-oriented programming: procedural approach Vs object oriented approach, principles of OOP: encapsulation, inheritance and polymorphism. Concepts of OOP: Abstraction, overloading, reusability, extensibility, dynamic binding, message passing. Introduction to Java: Importance and features of java, data types, Operators and expressions, Control statements. Introducing classes, objects and methods: defining a class, adding variables and methods, creating objects, constructors. Arrays and strings: creating an array, one and two dimensional arrays, string array and methods, String and String Buffer classes. **(12)**

UNIT II: Inheritance: Basics types, using super, Multilevel hierarchy, abstract and final classes. Using super - Method overriding - Dynamic Method Dispatch, Abstract keyword Using final with inheritance - the Object Class. Packages and interfaces: Basics, Access protection, Extending Interfaces and packages. CLASSPATH variable Exception Handling: Fundamentals, exception types, uncaught exceptions, try-catch block, throw, throws, finally, built in exceptions, creating your own exceptions **(12)**

UNIT III: Multithreaded Programming: Fundamentals, single and multiple thread creation, priorities, synchronization, Thread class, Thread class methods, Runnable interface, inter thread Communication, suspending, resuming and stopping threads. The Collections Framework, Collections Overview, Collection Interface, List Interface, Set Interface, SortedSet Interface, Queue Interface, Java Map Interface. ArrayList Class, LinkedList Class, HashSet Class, Using an Iterator – The For Each Statement. **(12)**

UNIT IV: Introduction to HIBERNATE - Advantages of Hibernate compared to JDBC, ORM (Object Relational Mapping) –Introduction to HIBERNATE Resources - Configuration File -Mapping File -Persistent Class Or POJO -Client Application Hibernate Architecture -Hibernate Architecture -Installation And Directory Structure - Installation And Directory Structure -Hibernate Data Types -Hibernate Data Types. Hibernate API - Configuration -Sessionfactory -Session –Transaction. Object Lifecycle In Hibernate - Transient Object -Persistent Object -Detached Object Crud Operations Using Session Methods -Save, Persist, Save or update, Update, Merge, Delete,Load, Get. **(12)**

UNIT V: Servlets: Servlets Basics – Life Cycle of a Servlet –The Servlet API – Servlet Interfaces – Generic Servlet ClassHttpServletRequest Interface – HttpServeltResponse Interface – HttpServet Class – The Cookie Class – Handling HTTP GET Request – Handling HTTP POST Request Spring Core - Introduction to Spring Frameworks –What is a Core Container –Introduction to IOC –Types of DI –Setter DI vs Constructor DI –Bean Scopes. **(12)**

Suggested List of Lab Exercises:

Implementation of the following OOPs and Java programming concepts learnt in theory.

- Class and objects
- Constructor
- Arrays (Single dimensional and Two Dimensional)
- String manipulation
- Inheritance-Different types of inheritance
- Abstract and final class
- Method overriding
- Packages
- Interfaces
- Exception handling
- Multithreading
- The Collection Framework
- Hibernate-CRUD operations using session methods
- Servlet

Reference Books:

1. Patrick Naughton and HerbertzSchildt, "Java-2 The complete Reference", 12th Edition, 2020.
2. Richard A. Johnson, "An Introduction to Java Programming and Object Oriented Application Development", CENGAGE Learning India Pvt. Ltd., New Delhi, 7th Edition, 2016.
3. E. Balagurusamy, "Programming with Java: A Primer", McGraw-Hill Education (India), New Delhi, 6th Edition, 2019.
4. Harvey M. Deitel & Paul J. Deitel, "Java: How to Program", PHI Learning Pvt. Ltd., New Delhi, 11th Edition, 2014.
5. Dr. R. Nageswara Rao, "Core Java An Integrated Approach", Dreamtech Press (India) Pvt. Ltd., Hyderabad, 2nd Edition, 2015.

22MCA106 PROBLEM SOLVING TECHNIQUES

Total No. of Hours: 60 **Hours per week: L:T:P (0:0:4)**

Course Objectives:

- To make the student learn different data structure for efficient problem solving.
- To demonstrate a familiarity with major algorithms and data structures.
- To synthesize efficient algorithms in common design situations.

Learning Outcomes:

On completion of this course, the student will be able to:

- Ability to design efficient algorithms using various algorithm designing strategies.
- Ability to classify the problem and apply the appropriate design strategy to develop algorithm.

Topics/Programs Covered:

1) Sorting problem and its usage in various applications.
2) Search techniques and its applications.
3) Program to implement Towers of Hanoi.
4) Program to implement Strassen's matrix multiplication using Divide and Conquer Technique.
5) Program to implement Prim's Algorithm using Greedy Technique.
6) Program to implement Kruskal's Algorithm using Greedy Technique.
7) Dijkstra's Algorithm to obtain the shortest paths.
8) Program to implement Travelling Salesperson Problem using Dynamic Programming Technique.
9) Program for Tree Traversals (preorder, inorder, postorder).
10) Working with approximation algorithm – Vertex cover problem

MANDATORY COURSE (SEMESTER I)

Foundations of Computer Science is a non-credit course offered in semester I to the students of non-computer science background.

BRIDGE COURSE: FOUNDATIONS OF COMPUTER SCIENCE

Total No. of Hours: 30 + 30

Course Objectives: The aim of the course is to introduce computer science concepts to students from non-computer science students and equip them with the current technologies that are used in the software development industry.

Learning Outcomes:

- To introduce the fundamental concepts of computers and computing environment.
- To acquire the basic knowledge of algorithm design and problem solving using C language.
- To get an overall idea of different technologies, tools & software's which are used in real time projects

UNIT – I: Digital computers and Digital system: Number systems, Number base conversion, Complements, Binary codes, Binary arithmetic's. Boolean algebra: definitions, Basic theorems and properties of Boolean algebra, Venn diagram. Introduction to Computer architecture – arithmetic, logic and control units, addressing modes. **(6)**

UNIT – II: Problem solving techniques: Introduction, Problem solving procedure. Algorithm: Steps involved in algorithm development, Algorithms for simple problems, Flowcharts, Psuedocode. **(4)**

UNIT – III: Introduction to C: Overview of C Program, Basic structure of a C - program. Constants, Variables & Data types: Character set, C token, Keywords & identifiers. Control Statements. Working with Arrays and Strings. **(8)**

UNIT – IV: Working with functions, recursive functions, working with pointers. **(6)**

UNIT – V: Overview of recent technologies: Introduction to Hibernate, Spring Framework, Angular Framework, Web services, Web Server, Code repositories, Build Tools - ANT, Maven, Jenkins - CI/CD, Cloud Services. **(6)**

Suggested List of Lab Exercises:

Demonstration and implementation of C programming concepts learnt in Theory.

- Using arithmetic expression
- Branching
- Looping

- Nesting of loops
- Arrays (Single dimensional and Two Dimensional)
- String manipulation
- Functions
- Recursion
- Pointers

Reference Books:

1. E. Balagurusamy, "Programming in ANSI C", 7th Edition, Tata McGraw Hill.
2. ITL Education Solutions Ltd, "Introduction to Information Technology", Pearson Education India, 2nd Edition, 2012.
3. K.R. Venugopal and Sudeep R Prasad, Programming with C, 4th Edition, Tata McGraw-Hill Education.
4. Yashavant P. Kanetkar, Let Us C, 10th Edition, Tata McGraw Hill, 2010.
5. M.Morris Mano, Digital Logic and Computer design, PHI, 2015.

Semester II

22MCA201 SOFTWARE ENGINEERING METHODOLOGIES

Total No. of Hours: 60 **Hours per week: L:T:P (3:1:0)**

Course Objectives: To give students a detailed understanding of processes and techniques for building large software systems, develop skills to evolve systems from analysis, to implementation, and to understand best ways of building software.

Learning Outcomes

After completion of this course, the student will be able to:

- Analyze a software problem
- Prepare Requirement analysis
- Preparing Test cases and perform Testing

UNIT I: Introduction to Software Engineering: Software Engineering development, Software Life Cycle Models, Comparison of various models, Introduction to Agile Principles and Scrum **(12)**

UNIT II: Requirements in various methodologies: Introduction to Object Oriented Methodology, Overview of Requirements Elicitation, Requirements Model, Action and Use cases, Requirements Elicitation Activities, Managing Requirements Elicitation, Requirements management in Agile Projects, Role of Product owner and Product backlog creation and grooming. **(12)**

UNIT III: Clean code – Introduction - Overview of clean code principles - Refactoring and code smells - differentiate good and bad code, Introduction to Design patterns: Clean code patterns - functions - comments - formatting - objects and data structures - error handling. **(12)**

UNIT IV: Advanced Software Engineering: Component-based Software Engineering: Components and component models, CBSE processes, Component composition; Distributed Software Engineering: Distributed systems, Client–server computing, SaaS. **(12)**

UNIT V: Software Testing, Project Planning and Management: - Software Testing: Development testing, Test driven development, Release testing, User testing; Project Planning and Management: Risk management, Plan-driven development, Project scheduling, Estimation techniques. **(12)**

References Books:

1. Ian Summerville, "Software Engineering", Pearson Education Ltd, 10th Edition, 2015, ISBN: 9780133943030.
2. Roger S Pressman, Bruce R. Maxim, "Software Engineering: A Practitioner's Approach", McGraw-Hill, 8th Edition, 2015, ISBN: 9780078022128.

3. "Guide to the Software Engineering body of Knowledge", Version 3.0 SWEBOK, IEEE Computer Society Press, 2014, E-Book: ISBN: 9780769551661.
4. Pankaj Jalote, "An Integrated Approach to Software Engineering", 3rd Edition, Narosa Publishing House 2013, ISBN: 81-7319-702-4.
5. Ken Schwaber, MikeBeedle, "Agile Software Development with Scrum", Pearson India, 2014.
6. Robert C.Martin, "Clean Code - A Handbook of Agile Software Craftsmanship", Prentice Hall, 2008.

Recommended Reading:

1. The Phoenix Project: A novel about IT, DevOps, and helping your Business Win-Gene Kim, Kevin Behr and George Spafford, It Revolution Press, 2018.

22MCA202 COMPUTER NETWORKS

Total No. of Hours: 60 **Hours per week: L:T:P(4:0:0)**

Course Objectives: This course provides the broad coverage of data communication system and computer networks. The students will be equipped with the knowledge about communication system components, internetworking, network topology, protocols, and algorithms.

Learning Outcomes

After completion of this course, the student will be able to:

- Understand the concept of networks, different topologies and network devices
- Discuss the objectives and functionalities of different layers.
- Describe how the available methods and algorithms are implemented in the real-time networks such as Ethernet, Bluetooth and internet protocols.
- Understand the working of few application protocols such as SMTP, POP and HTTP.

UNIT I: Introduction - Protocol and Standards - Hierarchies, Network Models, Layered Tasks - OSI Reference Model, Introduction to internetworking, TCP/IP Model.

Data Communication: Analog and Digital Signals, Digital Transmission – Line Coding, Block Coding, Sampling and Transmission Mode, Modulation Techniques, Networking Devices – hubs, switches, bridges, routers and gateways, Bandwidth utilization - Multiplexing and Spreading, Switching Techniques, Packet switching protocols, Introduction to X.25. **(12)**

UNIT II: Optical Networking – Introduction, SONET / SDH Standard, WAN protocol – Introduction to ATM, basic concepts of ATM Networking, ATM cells.
LAN Protocol Architecture – Wireless LAN's, Bluetooth, and High-Speed LAN's.
(12)

UNIT III: Internet and Transport Protocols **-** Internet basics, IP Protocol, Introduction to Intra-domain and inter-domain routings, Unicast Routing Protocols, Multicast Routing Protocols, Overview of RIP, OSPF, MOSPF and BGP protocols, Transport protocols - TCP, UDP protocols. **(12)**

UNIT IV: Software Defined Networking: Evolution of Switches and Control Planes, cost, Fundamental characteristics, operations, SDN devices, SDN controller, Openflow Specification: Openflow overview, OpenFlow 1.0 and 1.1, Introduction to Network Functions Virtualization, SDN vs NFV.
(12)

UNIT V: Network Security: Security Services, Message Confidentiality – Symmetric Key Cryptography, AES algorithm, Public Key cryptography, RSA algorithm, Message Integrity – hashing, SHA algorithm, MAC, Digital Signature, Entity Authentication, Key Management – KDC, Diffie Hellman and Kerberos. **(12)**

Reference Books:

1. Andrew S. Tanenbaum, "Computer Networks", Sixth Edition, Pearson Education, 2021.
2. Behrouz A. Forouzan, "Data communications and Networking", Tata McGraw-Hill, Fourth Edition, 2017.
3. Stallings William, "Data & Computer Communications", Pearson Education Asia, Tenth Edition, 2017.
4. Paul Göransson, Chuck Black, "Software Defined Networks, A Comprehensive Approach", 2nd Edition, 2016, Elsevier, ISBN: 978-0-12-416675-2
5. William Stallings, "Network Security Essentials", 6th edition, Pearson, 2017.
6. Dayanand Ambawade, Deven Shah, Mahendra Mehra, Mayank Agarwal, "Advanced Computer Network", Wiley Publications, 2011.

For Continuous Internal Assessment Only

Implementation of Computer Networking concepts such as Packet Capture and Analysis, Secure Communication Exercise, Network Vulnerability Assessment and Network troubleshooting using Cryptool / Packet Tracer / Similar tools.

23MCA204 WEB TECHNOLOGIES

Total No. of Hours: 60 + 60 **Hours per week: L:T:P (4:0:4)**

Course Objectives: This course aims to familiarize students with cutting-edge technologies that facilitate the development of intricate web applications. Participants will explore advanced JavaScript concepts for creating interactive web applications with sophisticated user interfaces. Additionally, they will delve into server-side web technologies that power dynamic web applications. The course also covers the creation of contemporary web applications using React and Node frameworks.

Learning Outcomes:

- Understand the foundations of web development and the client-server architecture.
- Create well-structured HTML documents and apply CSS styles for web design.
- Apply JavaScript effectively to create interactive and dynamic websites.
- Develop dynamic web applications using the React framework, including components, services, and routing.
- Demonstrate proficiency in developing and deploying a web application using React.js and Node.js.

Unit I: Basic Web Concepts: Introduction to World Wide Web (WWW), Web Browsers and Web Servers, Security and Vulnerability in Web Systems, Web System Architecture – URL and Domain Name, Client-side and Server-side Scripting, Web hosting options and deployment strategies, Web protocols – HTTP Request Response Format, Features of HTTP/1, HTTP/2 and HTTP/3 protocols.
HTML Fundamentals – Elements and attributes, Features of Web 3.0, HTML vs HTML5, Semantic HTML and Accessibility. CSS3.0 – Styles and Style sheets, Selectors, Style properties, Box Model, and Responsive design. **(12)**

Unit II: JavaScript: Introduction, Operators and Expressions, JavaScript Programming Constructs, Document Object Model, Working With Objects, Handling Events, Creating Frames and Windows, Processing Forms, Using Images, Regular expressions, Working with cookies. Introduction to JSON-JSON vs XML-JSON Objects, AJAX: Ajax Client Server Architecture - XML Http Request Object - Call Back Methods. ES6 and modern JavaScript features, Call-backs and Promises, Single Page Application, Asynchronous Communication. **(14)**

Unit III: React.js: React installation and application setup, JSX, React Classes and Components, Rendering of elements, Properties, State, Context, Component lifecycle methods, Refs & Keys, Handling events and forms, React Router, Stateless components, React form & controls, State management with Redux or Context API, Asynchronous data fetching with Axios or Fetch API, React hooks – useState, useEffect. **(12)**

Unit IV: Node.js: Understanding Node.js Architecture, Setting up a Node.js development environment, Working with modules and npm, Handling HTTP requests and responses with Node.js, query string, call backs, buffers, streams, File system, Working with databases – Simple SQL operations using MySQL: Reading and Writing to MySQL. Introduction to Testing frameworks Jest, Mocha, and Chai. Client-side Vs Sever-side rendering. **(12)**

Unit V : Express.js; Introduction to Web services and REST API's , Express Installation and Server setup, Building the application stack, Routing, List API, Create API, Error Handling, Express Middleware, User authentication and authorization – Types of Authentications in Node.js – JWT Token and OAuth, Express Scaffolding and Templates. **(10)**

Suggested List of Lab Exercises:

- Web page creation using HTML5 elements
- Using style sheets (CSS2 and CSS3)
- Client Side Scripting using Java Script
- Working with images, JavaScript objects and event handling
- Form validation using JavaScript, Working with JSON
- Designing webpages with React.js components
- State management with Redux or Context API,
- Asynchronous data fetching with Axios or Fetch API
- React hooks – useState, useEffect.
- Database handling using Node.js & MySQL
- Creating Simple dynamic web application.

Reference Books:

1. Laura Lemay, Rafe Colburn & Jennifer Kyrnin, "Mastering HTML, CSS & Javascript Web Publishing", BPB Publications, First edition,2016.
2. Harvey M Deitel, Paul J Deitel and Tem R Nieto, Internet and World Wide Web How to Program, Pearson, 6th Edition, 2020.
3. Minnick, C. Beginning ReactJS foundations building user interfaces with ReactJS: An Approachable Guide, OReillly, 2022.
4. Chris Northwood, The Full Stack Developer: Your Essential Guide to the Everyday Skills Expected of a Modern Full Stack Web Developer, Apress Publications, 1st Edition, 2018.
5. Brown, Ethan, "Web Development with Node and Express: Leveraging the JavaScript Stack", O'Reilly Media, Second Edition, 2019.
6. "Learning React, Functional Web Development with React and Redux", Alex Banks and Eve Porcello, O'Reilly Media, May 2017.

22MCA205 MOBILE APPLICATION DEVELOPMENT

Total No. of Hours: 60 + 60 **Hours per week: L:T:P (4:0:4)**

Course Objectives: This course aimed at helping students to learn about mobile applications. It helps students to learn about Android operating system and also to develop applications on Android open-source platform. The course explains the setup of the Android Studio-based development tools, the Android SDK, all essential features, as well as the advanced capabilities and APIs

Learning Outcomes

After completion of this course, the student will be able to:

- Design activities, Fragments
- Sending email , messages through app
- Develop Apps for various applications

UNIT I: Introduction to Android

An open platform for mobile application development(the android ecosystem) - android sdk features - introduction the development framework - Android software stack - Android Runtime - Android application architecture(Android components) - android application lifecycle - application manifest and permissions. **(12)**

UNIT II: Activities and Android User Interface

Understanding Activities - Activity life cycle - Understanding Fragments - Fragment Lifecycle - Views, Viewgroup and Layouts - commonly used layouts(linear layout, constraint layout, frame layout) - linking activities and fragment with layout views - Intents - calling built in application using intents - Navigation between activities and fragments - Understanding components of screen - Adapting to display and screen orientation using toolbar. **(14)**

UNIT III: Designing with views and displaying pictures and menus

Basic android views - create user interface using basic views and layouts using xml and programmatically - handling user interaction events - Adapters and using listview and recycler view to show display list - using picker views - using image views to display pictures - using menus with views -using webview - UI notifications - fragment transaction, adding and removing fragments. **(12)**

UNIT IV: Data persistence and Content providers

Storage overview - shared preference - saving and loading user preference - file system - saving and retrieving files - introduction android databases - Sqlite database and content providers - working with Sqlite databases (SQLiteOpenHelper, android database design considerations, CRUD operations) - creating and using content providers. **(12)**

UNIT- V: Messaging, Location based services and Networking

SMS messaging - Sending Email - Consuming web services using HTTP using basic android apis and retrofit library.

Introduction to cross platform framework for hybrid mobile development. **(10)**

Suggested List of Lab Exercises:

Demonstration and implementation of features of Android

1. Implementation of the features activities, intent and fragments.
2. User Interface and Designing with views.
3. Utilizing the Action Bar.
4. Listening for UI Notifications.
5. Implementation of list views to display list.
6. Usage of specialized fragments.
7. Implementation of content providers.
8. Using image views to display pictures.
9. Implementation of menus with views.
10. Implementation of CRUD operations using SQLite

In addition, the students will develop an Android App using the concepts Learned.

Reference Books:

1. Reto Meier, "Professional Android", 4th Edition, Wiley Publications, 2018.
2. John Horton, "Android Programming for Beginners", 2nd Edition, Ingram short title, 2018.
3. John Paul Mueller and Barry Burd, "Android Application Development All-In-One for Dummies", 3rd Edition, Wiley, 2021.
4. Aleksei Sedunov, "Kotlin In-Depth: A Guide to a Multipurpose Programming Language for Server-Side, Front-End, Android, and Multiplatform Mobile", 3rd Edition, BPB Publications, 2022.
5. Shaun Lewis, Mike Dunn, "Native Mobile Development", O'Reilly Media, 2019.

22MCA206 EXPLORING PYTHON

Total No. of Hours: 60 **Hours per week: L:T:P (0:0:4)**

Course Objectives:
This course introduces the student to Python programming language through its core language basics and program design techniques suitable for modern applications.
To understand the wide range of programming facilities available in Python and to utilize high-performance programming constructs to develop solutions in real life scenarios.

Learning Outcomes:
After completion of this course, the student will be able to:

- To understand the wide range of programming facilities available in Python and develop solutions to simple computational problems
- To utilize high-performance programming constructs available in Python to represent and process compound data
- To develop solutions for real-time projects in areas that includes data analytics, visualization, image processing, artificial intelligence and machine learning

Topics / Programs Covered:

- Conditional and looping constructs, Functions
- Modules - Functions from math, random, time & date module.
- Strings: Regular Expressions and Pattern matching
- List – Accessing List Elements, List operations, List comprehensions
- Dictionaries: Traversing, appending, updating and deleting elements
- Tuples
- I/O and File Handling – Working with CSV files and other file formats
- Arrays and Matrices – Working with NumPy Module
- Object Oriented Concepts - Python Scopes and Namespaces, Inheritance
- Working with Pandas - Data indexing, selection, Operating on Data
- Creating simple 2D and 3D plots using Matplotlib

Reference Books:

1. Paul Gries, Jennifer Campbell, Jason Montojo, “Practical Programming: An introduction to Computer Science Using Python”, 2nd Edition, The Pragmatic Bookshelf, 2013.
2. Allen Downey, Jeffrey Elkner, “Learning with Python: How to Think Like a Computer Scientist”, Createspace Independent Publishers, 2015.
3. Hans Fangohr, “Introduction to Python for Computational Science and Engineering (A beginner's guide)”, 2015.
4. Timothy A. Budd, “Exploring Python”, Mc Graw Hill Education, 2011.
5. Mark Lutz, “Learning Python”, 4th Edition, O’Reilly publication, 2012.

MANDATORY COURSE (SEMESTER II) - MOOC COURSE

Course Objectives: The goal of this course is to motivate the students to gain knowledge in any emerging technology that is not part of the regular academic curriculum. This exposure will make them outstanding while facing real competition.

The student has to choose a MOOC course of 30 -45 hours duration in consultation and guidance from their mentor (faculty-adviser). The final assessment certificate has to be submitted to complete the course.

Suggested MOOC platforms:

- SWAYAM / NPTEL
- Coursera
- AWS Academy

MANDATORY COURSE (SEMESTER III) - PLACEMENT TRAINING

Course Objectives: The goal of this course is to help the students to explore the relationship between the knowledge & skills acquired in college with those required in the working situations.

Quantitative aptitude and reasoning - Work ethics - Business email writing - interview skills - Group Discussions - Communication skills - Assertive Techniques and confidence building

MANDATORY COURSE (SEMESTER IV) – RESEARCH PAPER PRESENTATION / PUBLICATION

Course Objectives: The goal of this course is to encourage the students to identify innovative research problems that help in exploring variables that promote creativity and innovation. Each student is expected to choose an area of research of their interest, preferably the research work carried out in Semester III and must publish at least one (1) research paper in refereed journal / make presentations in conferences before the end of the semester and produce evidence for the same in the form of presentation certificates and/or reprints.

Discipline Specific Electives

22MCA203A ARTIFICIAL INTELLIGENCE

Total No. of Hours: 60 **Hours per week: L:T:P (3:1:0)**

Course Objectives:
The objective of the paper is teach modern view of AI and its foundation. It illustrates search strategies with algorithms and problems. It teaches different techniques of NLP and Game Playing.

Learning Outcomes:

- Identify problems that are amenable to solution by AI methods.
- Identify appropriate AI methods to solve a given problem
- Formalize a given problem in the language/framework of different AI methods
- Implement basic AI algorithms
- Design and carry out an empirical evaluation of different algorithms on a problem formalization, and state the conclusions that the evaluation supports

UNIT I: Introduction to AI - The Foundations of AI, AI Technique -Tic-Tac-Toe. Problem characteristics, Production system characteristics, Production systems: 8-puzzle problem. Searching: Uniformed search strategies – Breadth first search, depth first search.
Local Search Algorithms: Generate and Test, Hill climbing, simulated annealing search, Constraint satisfaction problems, Greedy best first search, A* search, AO* search. **(12)**

UNIT II: Game Playing Overview, Minimax algorithm, Alpha-Beta pruning, Additional Refinements. Probabilistic Reasoning: Ad Hoc Methods. **(8)**

UNIT III: Knowledge-based agents; Logic: propositional logic reasoning patterns in propositional logic; Effective propositional inference; Agents based on propositional logic. Representation revisited; Syntax and semantics of first-order logic; Using first-order logic; Knowledge engineering in first-order logic. Propositional versus first-order inference; Unification and lifting, Forward chaining; Backward chaining; Resolution, Truth maintenance systems. **(14)**

UNIT IV: Basic plan generation systems – Strips -Advanced plan generation systems – Kstrips – Strategic explanations -Why, Why not and how explanations. Learning: Forms of Learning; Inductive learning; Learning decision trees; Ensemble learning; Computational learning theory. Handling Uncertainties: Non-monotonic reasoning, Probabilistic reasoning, use of certainty factors, Fuzzy logic. **(14)**

UNIT V: Computer Vision, Natural Language Processing: Introduction, Syntactic Processing, Semantic Analysis, Discourse and Pragmatic Processing, Spell Checking.

Expert Systems: Need and justification for expert systems, Architecture and role of expert systems, Case studies: MYCIN **(12)**

Reference Books:

1. E. Rich and K. Knight, “Artificial Intelligence”, 3rd Edition. New York: TMH, 2019.
2. S. Russell and P. Norvig, “Artificial Intelligence A Modern Approach”, 4th Edition. Pearson Education, 2020.
3. Andries P. Engelbrecht, "Computational Intelligence: An Introduction", John Wiley & Sons, 2nd Edition, 2009.
4. Eugene Charniak and Drew McDermott, “Introduction to Artificial Intelligence”, 2nd Edition, Pearson Education, 2009.
5. George F Luger, “Artificial Intelligence Structures and Strategies for Complex Problem Solving”, 6th Edition. Pearson Education, 2009.

For Continuous Internal Assessment Only

Implementation of basic algorithms learnt in the theory.

22MCA203B INTERNET OF THINGS (IoT)

Total No. of Hours: 60 **Hours per week: L:T:P (3:1:0)**

Course Objectives:

- Assess the genesis of IoT applications, architectures in real world.
- Illustrate diverse methods of deploying smart objects and connect them to network.
- Compare different Application protocols for IoT.
- Infer the role of Data Analytics and Security in IoT.
- To build simple IoT Systems using Arduino.

Learning Outcomes:

- Interpret the impact and challenges posed by IoT networks.
- Analyze various protocols for IoT.
- Gain knowledge on various IoT Platforms.
- Exposure to sensors and to design IoT system using Arduino.

UNIT I: **Introduction to IoT:** Genesis of IoT, IoT and Digitization, Convergence of IT and OT, IoT Challenges.
IoT Network Architecture and Design: The one M2M IoT standardized architecture, The IoT World Forum (IoTWF) standardized architecture. A simplified IoT Architecture – The core IoT functional stack. IoT data management and complete stack. **(12)**

UNIT II: Smart Object: The "Things" in IoT – Sensors, Actuators, and Smart Objects. Sensor Networks – Wireless Sensor Networks, Communication protocols for WSNs.
Connecting Smart Objects: Communication criteria, IoT access Technologies – IEEE 802.15.4, IEEE 1901.2a, IEEE 802.11ah, LoRaWAN. **(12)**

UNIT III: IoT Protocols: Internet Protocol – The key advantages of IP, Adoption or Adaptation of IP, The need for optimization, Optimizing IP for IoT – from 6LoWPAN to 6Lo, Header compression, Fragmentation, Mesh addressing, 6TiSCH, RPL.
Application protocols for IoT – The transport layer, IoT Application Transport methods – Adapting SCADA for IP, IoT Application layer protocols – COAP, MQTT. **(12)**

UNIT IV: Data Analysis for IoT: An Introduction to data analytics for IoT, Machine learning, Big data analytics tools and technology – NoSQL database, Hadoop Ecosystem, Apache Kafka, Lambda architecture. Edge streaming analytics – Edge analytics core function.
IoT Security – challenges in IoT security, Risk analysis structures – OCTAVE and FAIR. **(12)**

UNIT V: IoT Platforms: Working with IoT Platforms – Arduino UNO, Integrate Arduino with Thingspeak, Temboo, Interfacing Arduino and Blynk via USB and ESP8266 – LED Blinking.
Industrial IOT: Case Study – Smart and Connected Cities, Transportation. **(12)**

Reference Books:

1. David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Robert Barton, Jerome Henry, "IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things", 1stEdition, Pearson Education (Cisco Press Indian Reprint). (ISBN: 9789386873743)
2. Srinivasa K G, "Internet of Things", CENGAGE Leaning India, 2017.
3. Olivier Hersent, David Boswarthick, Omar Elloumi , ―The Internet of Things – Key applications and Protocols‖, Wiley, 2012.
4. Michael Margolis, Arduino Cookbook, Recipes to Begin, Expand, and Enhance Your Projects, 2nd Edition, O'Reilly Media, 2011.

For Continuous Internal Assessment Only
Implementation of simple IoT applications with Arduino board and basic sensors.

22MCA203C DIGITAL IMAGE PROCESSING

Total No. of Hours: 60 **Hours per week: L:T:P (3:1:0)**

.

Course Objectives: This course exposes the students to the basics of digital image processing and its applications. It covers a broad range of image processing techniques such as Image enhancement, restoration, segmentation and feature analysis. The course also provides develop on-hand experience in applying these techniques to process the images.

Learning Outcomes

- Have a clear perceptive and practical scope of digital image processing, current technologies and issues that are specific to image processing systems.
- Understand the working of different image processing algorithms such as filtering, segmentation, morphological processing and image representation.
- Could implement basic image processing algorithms using image processing tools such as MATLAB or OpenCV

UNIT I: Fundamentals of Image Processing: Introduction, Components of image processing system, Elements of visual perception, Steps in Image Processing Systems, Image Acquisition, Sampling and Quantization, Pixel Relationships, Colour Fundamentals and RGB, CMY, HSV colour Models.
Mathematical Preliminaries: Vector algebra, Matrix operations, Fourier Transform. **(12)**

UNIT II: Image Enhancement: Introduction to Spatial and Frequency domain, Image Operations, Arithmetic, Logical, Statistical and Spatial Operations, Convolution and Correlation, Enhancement in Spatial Domain – Gray level Transformations, Histogram Processing, Spatial Filtering, Smoothing and Sharpening. **(12)**

UNIT III: Filtering in Frequency Domain: Smoothing and Sharpening filters, Homomorphic Filtering.
Image Restoration: Noise models, Constrained and Unconstrained restoration models. **(10)**

UNIT IV: Image Segmentation: Detection of Discontinuities, Edge Operators, Edge Linking and Boundary Detection, Thresholding, Region Based Segmentation.
Feature Analysis and Extraction: Image Features, Textures, Boundary representations and Descriptions, Component Labeling, Regional descriptors and Feature Selection Techniques. **(14)**

UNIT V: Image Morphology: Binary and Gray level morphology operations, Erosion, Dilation, Opening and closing operations, Distance transforms, Basic morphological operations.

Image Compression: Error criterion, Lossy Compression, Lossless Compression, Huffman Coding, Run length Coding, Block Coding, Quad Tree Coding, contour Coding. . **(12)**

Reference Books:

1. Rafael C.Gonzalez and Richard E.Woods, "Digital Image Processing", Third Edition, Pearson Education, 2008.
2. S. Sridhar, "Digital Image Processing", Oxford University Press, 2011.
3. Jayaraman S, Veerakumar T, Esakkirajan S, "Digital Image Processing", Tata McGraw-Hill Education, 2011.
4. Rafael C. Gonzalez, Richard E. Woods , Steven L. Eddins ,"Digital Image Processing Using Matlab 2E", Gatesmark Publishing, 2009.

For Continuous Internal Assessment Only

Implementation of image processing concepts such as enhancement, restoration, segmentation and compression using Matlab / ImageJ / OpenCV / Similar tools

TEACHING-LEARNING PROCESS

The course of study is organized on a semester basis consisting of 4 semesters in 2 years. The rigors of academic study at each level are balanced with a number of related activities. Various activities are framed to reach the goals of the course from the first semester to the last semester. Classroom learning is made interesting through presentations, simulation, assignments, group discussion and projects. Experts from industry, alumni and academia supplement formal classroom sessions with frequent guest lectures. National level Conferences, Seminars and Workshops are frequently arranged to update the students in the emerging IT trends and to ensure that students are placed in top companies in the industry.

The first year of study emphasizes on the problem solving skills and equip them with strong knowledge in fundamentals. To develop a holistic view of education focusing on the intellectual, emotional, ethical and social aspects, value education is provided to the students of all the semesters. The subjects in the third semester provide a modern, industry-oriented education in applied computer science so that the students can excel in the present-day global situation. 30 hrs. of communication skills development course and 30 hrs. of personality development programmer conducted enhances their soft skills and prepare them to face the corporate world.

Subjects in the third and fourth semesters offer a creative atmosphere in higher studies and research amongst the students. Research methodology and Research-based project or industry work course motivate the students towards research. The students are encouraged to publish and present their research work. Seminars and workshops are conducted throughout the year to improve their technical skills and make them industry-ready. Throughout the course, students are equipped with programming, analytical ability, strong knowledge with applications of theory. This enhances their prospects for a career in various fields.

Teaching-Learning Methodologies Adopted

- Interactive Lectures & Discussions
- Presentations through Slides, Video
- Demonstrations & Simulations
- Case studies
- Project work
- Group discussions
- Corporate Sessions
- Sessions by Alumni
- Industry visits
- Sharing of work experience / knowledge
- MOOC Courses
- Journals