

# MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE

**Madanapalle**

(UGC-AUTONOMOUS)

[www.mits.ac.in](http://www.mits.ac.in)



## DEPARTMENT OF COMPUTER APPLICATIONS

**Course Structure**

**And**

**Detailed Syllabus (R22)**

**For the students admitted to**

**Master of Computer Applications from the academic year 2022-23 batch onwards**



### MCA I Year I Semester

S.No	Course Code	Course Title	L	T	P	C
1	22MATP101	Mathematical Foundations for Computer Applications	3	1	0	4
2	22MCAP101	Python Programming	3	1	0	4
3	22MCAP102	Database Management Systems	3	1	0	4
4	22MCAP103	Computer Organization and Architecture	3	0	0	3
5	22MCAP104	Operating Systems	4	0	0	4
6	22MCAP105	Computer Networks	4	0	0	4
7	22ENGP201	Corporate Communication Laboratory	0	0	4	2
8	22MCAP201	Python Programming Laboratory	0	0	3	1.5
9	22MCAP202	Database Management Systems Laboratory	0	0	3	1.5
10		Skill Oriented Course – I	1	0	2	2
<b>Total</b>			<b>21</b>	<b>3</b>	<b>12</b>	<b>30</b>

### MCA I Year II Semester

S.No	Course Code	Course Title	L	T	P	C
1	22MCAP106	Full Stack Web Development	4	0	0	4
2	22MCAP107	Data Structures and Algorithms	3	1	0	4
3	22MCAP108	Software Engineering	4	0	0	4
4	22MCAP109	Cryptography and Network Security	3	1	0	4
5	22MCAP110	Artificial Intelligence	4	0	0	4
6		Discipline Elective-I	3	0	0	3
7	22MCAP203	Full Stack Web Development Laboratory	0	0	3	1.5
8	22MCAP204	Data Structures and Algorithms Laboratory	0	0	3	1.5
9	22MCAP205	Software Engineering Laboratory	0	0	3	1.5
10		Skill Oriented Course – II	1	0	2	2
<b>Total</b>			<b>22</b>	<b>2</b>	<b>11</b>	<b>29.5</b>

**Tentative Course Structure for II Year MCA Programme:  
MCA II Year I Semester**

<b>S.No</b>	<b>Course Code</b>	<b>Course Title</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
1	22MCAP111	Mobile Application Development	4	0	0	4
2	22MCAP112	Fundamentals of Data Science	3	1	0	4
3	22MCAP113	Java Programming	3	1	0	4
4		Discipline Elective-II	3	0	0	3
5		Discipline Elective-III	3	0	0	3
6		Open Elective – I	3	0	0	3
7	22MCAP206	Mobile Application Development Laboratory	0	0	3	1.5
8	22MCAP207	Data Science using python Laboratory	0	0	3	1.5
9	22MCAP208	Java Programming Laboratory	0	0	3	1.5
10		Audit Course	2	0	0	0
11	22MCAP701	Mini Project	0	0	4	2
<b>Total</b>			<b>21</b>	<b>2</b>	<b>13</b>	<b>27.5</b>

**MCA II Year II Semester**

<b>S.No</b>	<b>Course Code</b>	<b>Course Title</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
1	22MCAP702	Comprehensive Viva	0	0	4	2
2	22MCAP703	Internship/Major Project	0	0	26	13
<b>Total</b>			<b>0</b>	<b>0</b>	<b>30</b>	<b>15</b>

## DISCIPLINE ELECTIVES

### Discipline Elective – I

S.No	Course Code	Course Title	L	T	P	C
1	22MCAP401	Machine Learning - Algorithms and Applications	3	0	0	3
2	22MCAP402	IoT Technology and Applications	3	0	0	3
3	22MCAP403	Agile Software Development Process	3	0	0	3
4	22MCAP404	Computer Graphics and Multimedia	3	0	0	3
5	22MCAP405	Image Processing	3	0	0	3
6	22MCAP406	Blockchain Technology	3	0	0	3
7	22MCAP407	XML and Web Services	3	0	0	3

### Discipline Elective – II

S.No	Course Code	Course Title	L	T	P	C
1	22MCAP408	Deep Learning	3	0	0	3
2	22MCAP409	Wireless Sensor Networks	3	0	0	3
3	22MCAP410	Software Quality Assurance and Testing	3	0	0	3
4	22MCAP411	Virtual Reality	3	0	0	3
5	22MCAP412	Video Analytics	3	0	0	3
6	22MCAP412	Software Development using Block chain	3	0	0	3
7	22MCAP413	Web Programming	3	0	0	3

### Discipline Elective – III

S.No	Course Code	Course Title	L	T	P	C
1	22MCAP414	Reinforcement Learning	3	0	0	3
2	22MCAP415	Privacy and Security in IoT	3	0	0	3
3	22MCAP416	Software Project Management	3	0	0	3
4	22MCAP417	Augmented Reality	3	0	0	3
5	22MCAP418	Computer Vision	3	0	0	3
6	22MCAP419	Information Security using Blockchain	3	0	0	3
7	22MCAP420	Computer Graphics and Multimedia	3	0	0	3

### OPEN ELECTIVE

#### Open Elective – I

S.No	Course Code	Course Title	L	T	P	C
1	22MBAP301	Management and Organizational Behavior	3	0	0	3
2	22MBAP302	Design Thinking	3	0	0	3
3	22MBAP303	Management Information Systems	3	0	0	3
4	22MBAP304	E-Commerce and Digital Markets	3	0	0	3
5	22MBAP305	Entrepreneurship Development and Project Management	3	0	0	3

**Skill – Oriented Course - I**

<b>S.No.</b>	<b>Course Code</b>	<b>Course Title</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
1	22MCAP601	Programming with C++	1	0	2	2
Any Skill-Oriented courses can be appended in future.						

**Skill – Oriented Course - II**

<b>S.No.</b>	<b>Course Code</b>	<b>Course Title</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
1	22MCAP602	Programming with Matlab	1	0	2	2
Any Skill-Oriented courses can be appended in future.						

## Audit Course

S.No.	Course Code	Course Title
1	22ENGP901	Soft Skill
Swayam NPTEL MOOCs ( Students ca opt to be assessed either in conventional mode or through proctored examinations conducted by Swayam NPTEL)		
2	22ENGP9M01/ 22ENGP9M01C	Soft Skills
3	22ENGP9M02/ 22ENGP9M02C	Developing Soft Skills and Personality
4	22ENGP9M03/ 22ENGP9M03C	Soft Skill Development
5	22ENGP9M04/ 22ENGP9M04C	Enhancing Soft Skills and Personality
Any Soft Skill related courses can be appended in future.		

# **MCA I Year I Semester**



## MCA I Year I Semester

### 22MATP101 MATHEMATICAL FOUNDATIONS FOR COMPUTE APPLICATIONS

L T P C  
3 1 0 4

**Course Prerequisite:** Basic Mathematics

#### **Course Description:**

This course introduces the concepts of set theory, relations, functions, logic and rules of inference. It illustrates the structured approach to graph theory models. It provides the valuable basic information on mathematical statistics. It explains about key concepts of probability and random variables and probability distributions.

#### **Course Objectives:**

Students will be able to:

1. Understand the mathematical reasoning to construct mathematical and logical arguments.
2. Understand various types of relations, Functions and posets.
3. Illustrate the different terminologies of graph theory and related concepts.
4. Identify the fundamentals of mathematical statistics.
5. Examine key concepts of probability, random variables and Probability distributions.

#### **UNIT-I PROPOSITIONAL LOGIC**

**9 hours**

Propositions, Connectives, Conditionals and Biconditionals, Well-formed formulae, Tautologies, Equivalence of formulas, Normal Forms, Rules of inference.

#### **UNIT-II RELATIONS, FUNCTIONS AND POSETS**

**9 hours**

Set Theory, Functions and Relations, Properties of a Relation, Binary relations, Matrix and Digraph representation of a relation, Equivalence relation, Partially ordered Relations, POSETS, Hasse Diagram, Lattices, Properties of Lattices.

#### **UNIT-III GRAPH THEORY**

**9 hours**

Basic terminology for undirected and directed graphs, multigraphs and weighted graphs, paths and circuits, Eulerian paths and circuits, Hamiltonian paths and circuits, Planar Graphs, Graph Coloring, Cut sets. Trees: Introduction to Trees, Tree terminology, Prefix codes.

#### **UNIT-IV DESCRIPTIVE STATISTICS**

**9 hours**

Measures of central tendencies and Dispersion, Coefficient of variation, Skewness, Kurtosis, Data visualization, Grouped data, Histograms, Ogives, Percentiles, Box-Plot, Correlation, Scatter diagram, Rank correlation and Linear Regression.

## **UNIT-V PROBABILITY**

**9 hours**

Introduction to Probability, , Axioms of probability, Conditional Probability, , Bayes theorem, Random Variable, discrete and continuous probability densities, cumulative distribution, Expectation, variance and standard deviation, Binomial distribution, Poisson Distribution and Normal Distribution.

### **Course Outcomes:**

After completion of the course the student will be able to

1. Form truth tables, prove results by truth tables, find normal forms and learn proof techniques and concepts of inference
2. Construct the matrix, digraphs of relation, Identify different types of relations, functions and draw the Hasse diagram for poset.
3. Apply graph theory models to solve different communication and network problems.
4. Analyze the data and characterize the nature of the distribution of data.
5. Solve the real time problems through the probability and its distributions.

### **Text Books:**

1. Kenneth H Rosen, Discrete Mathematics and Its Applications, 7th Edition, 2017, McGraw Hill Education.
2. Aczel Sounder Pandian, Complete Business Statistics, 7th Edition, McGraw Hill Education.

### **Reference Books:**

1. J.S. Milton and J.C. Arnold, Introduction to Probability and Statistics, 4<sup>th</sup> edition, 2003 Tata McGraw-Hill Publications.
2. C.L. Liu & Mohapatra, Elements of Discrete Mathematics, 4th Edition, 2017, McGraw Hill Education.
3. Thomas Koshy, Discrete Mathematics with Applications, 2012, Elsevier Academic Press.
4. S. C Gupta and V.K Kapoor, Fundamentals of Mathematical Statistics, 11<sup>th</sup> edition 2010 Sultan Chand & Sons, New Delhi.

**Mode of Evaluation:** Assignments, Internal Mid Examination, External End Examination

**MCA I Year I Semester**

**22MCAP101 PYTHON PROGRAMMING**

**L T P C**

**3 1 0 4**

**Course Prerequisite:** None

**Course Description:**

This course provides the students how to write programs in python language to perform different tasks in major concepts such as functions, collections, comprehensions, files, exception handling and OOPS.

**Course Objectives:**

Student will be able to

1. Understand the basics of python programming such as variables, data types, operators, etc.
2. Understand the concept of Lambda and user defined functions
3. Understand the use of collections such as string, list, set, tuple, dictionary
4. Understand the use of comprehensions and files
5. Understand the concepts of exceptions and OOPS

**UNIT-I FUNDAMENTALS 9 hours**

History, Python-Interpreter, Keywords, Variables, Datatypes, Operators, Expressions, Input and Output functions, Type conversion functions, Indentations, Conditionals - if constructs, Loop Structures - For loop, While loop, break and continue statements, Local and Global variables, Modules, Packages and Libraries

**UNIT-II LAMBDA AND USER DEFINED FUNCTIONS 9 hours**

Lambda functions, User Defined Function definition and function call, parameter passing, Pass and Return statements, Types of functions -Default arguments, Arbitrary arguments using \*, Keyword arguments, Arbitrary Keyword arguments using \*\*, Recursive functions.

**UNIT-III COLLECTIONS 9 hours**

String operations and functions, Slicing and Indexing operations, format function, multiple inputs using split function, List operations and functions, Tuple operations and functions, Set operations and functions, Dictionary operations and functions, Array and its operations

**UNIT-IV COMPREHENSIONS AND FILE HANDLING 9 hours**

List comprehensions - for, for with if , for with if and else, nested comprehensions with for, Dictionary comprehensions, Dictionary comprehension with if, Set comprehensions, map, zip, filter and other related functions, File path, opening and closing files, reading and writing files, File position.

**UNIT-V EXCEPTION HANDLING AND OOPS****9 hours**

Errors and Exceptions, Exception Handling, Multiple Exceptions, Raise statement, User Defined Exceptions, OOPS-Class, Object, Constructor, Methods, Inheritance and its types, super function, method overriding, Encapsulation, Polymorphism.

**Course Outcomes:**

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

1. Able to write simple programs in python
2. Able to write lambda and user defined functions
3. Write programs in collections - string, list, set, tuple, dictionary
4. Write programs using comprehensions and files
5. Write programs for exceptions and OOPS concepts

**Text Books:**

1. Python Programming – An Introduction to computer science, John Zelle, JimLeisy
2. Programming and Problem Solving with Python by Ashok Namdev Kamthane and Amit Ashok Kamthane, McGraw Hill Education; First edition (1 November 2017)

**Reference Books:**

1. Programming Python, Mark Lutz, O'Reilly, 3<sup>rd</sup> Edition, 2006
2. Core Python Programming, Wesley J Chun, PH, 2<sup>nd</sup> Edition Python Programming: A Compatible Guide for Beginners to Master and Become an Expert in python programming Language, Brain Draper, CreateSpace Independent Publishing Platform, 2016

**Mode of Evaluation:** Assignments, Internal Mid Examination, External End Examination

**MCA I Year I Semester**

**22MCAP102 DATABASE MANAGEMENT SYSTEMS**

<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
<b>3</b>	<b>1</b>	<b>0</b>	<b>4</b>

**Course Prerequisite:** None

**Course Description:**

This course is designed to investigate how database management system techniques are used to design, develop, implement and maintain database applications in organizations.

**Course Objectives:**

Students will be able to:

1. Explain data base concepts and its applications, data base system structure and ER data modeling.
2. Illustrate the relational model, DDL, DML and Relational Algebra.
3. Demonstrate Schema refinement and Normal Forms.
4. Explain transactions and its management, concurrency control schemes and recovery.
5. Interpret the concepts of PL/SQL and No SQL.

**UNIT I: INTRODUCTION**

**9 hours**

Introduction to data bases, Database system Vs File system, Data abstraction, Instances and Schemas, Database users, Database system structure, Entities, Attributes, Entity sets, Relationships and Relationship sets, Database design and ER diagrams, Specialization and Generalization and participation features of ER model.

**UNIT II: THE RELATIONAL MODEL, SQL & RELATIONAL ALGEBRA**

**9 hours**

**The Relational model:** Introduction, Various types of Integrity constraints.

**SQL:** Structured Query Language, data types, DDL, DML, TCL, views, NULL Values, Set operations, aggregate functions, character functions, Date functions, String functions, Nested queries, correlated nested queries.

**Relation Algebra:** Fundamental operations, Additional operations.

### **UNIT III: NORMALIZATION**

**9 hours**

Problems Caused by redundancy, FIRST, SECOND,THIRD Normal forms, BCNF, Lossy decomposition, Lossless-join Decomposition, Dependency preserving, Multi valued Dependencies , Fourth Normal Form and Fifth Normal form.

### **UNIT IV:TRANSACTION MANAGEMENT AND CONCURRENCY CONTROL 9hours**

**Transaction Management:** ACID properties, Transactions and Schedules, Concurrent Execution of transactions, Serializability, Conflict serializability, View serializability, Testing for serializability.

**Concurrency Control:** Two Phase Locking protocol, Time stamp ordering protocol

Database recovery

### **UNIT V: PL/SQL & NO SQL**

**9 hours**

**PL/SQL:** Functions, procedures, triggers, cursors and exceptional handling, Package in PL/SQL.

**No SQL :** Key value store Database, Column Store Database, Document Database, Graph Database

#### **Course Outcomes:**

After completion of the course the student will be able to

1. Demonstrate the basic concepts and advantages of DBMS and draw ER diagram for given problem.
2. Implement DDL, DML and various types of SQL queries.
3. Design Normal Forms for the given application.
4. Illustrate the concepts of transaction processing, concurrency control and recovery.
5. Execute a relational database system using PL/SQL and provide introduction of No SQL

#### **Text Books:**

1. Silberschatz, Korth, Sudarshan Data Base System Concepts, 6/e, TMH, 2019
2. Raghurama Krishnan, Johannes Gehrke, Data Base Management Systems, TMH
3. Sadalage Pramod J, Fowler Martin, NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot persistence , 1<sup>st</sup> Edition, Addison-Wesley, 2013

#### **Reference Books:**

1. Elmasri Navathe, Data Base Management System, Pearson publications
2. C J Date, An Introduction to Database Systems, Pearson

**Mode of Evaluation:** Assignments, Internal Mid Examination, External End Examination



## **UNIT V I/O ORGANIZATION**

**9 hours**

Input Output Interface, Connect different Input /Out Put devices, Asynchronous Data Transfer, Modes of Transfer, Priority interrupts, Direct Memory Access.

**Multiprocessor:** Characteristics of Multiprocessor, Inter Connection Structure, Inter Process Arbitration, Inter Process Communication and synchronization, cache Coherence.

### **Course Outcomes:**

After successful completion of the course, students will be able to

1. Demonstrate the basic components of Digital Computer.
2. Explain the internal working concept of processing unit along with instruction sets.
3. Identify the different Pipelining flow characteristics.
4. Illustrate various memory organization structures.
5. Analyze the I/O Organization Interface structure along with multiprocessor

### **Text Book(s)**

1. Mano, M. Morris. Computer system Architecture. Prentice-Hall of India, 4th Edition, 2003.
2. Stallings, W. Computer Organization and Architecture: designing for performance. Pearson Education India, 2003.

### **Reference Books**

1. Computer Organization, Carl Hamacher ,Zaky
2. Computer Architecture and Organization, Miles Murudocca, Vincent Heuring ,Wiley
3. Computer System Organization and Architecture, M.Usha, T.S.Srikanth

**Mode of Evaluation:** Assignments, Mid Term Tests, End Semester Examination.



**MCA I Year I Semester**

**22MCAP104 OPERATING SYSTEMS**

**L T P C**  
**4 0 0 4**

**Pre-requisite:** NIL

**Course Description:**

The purpose of this course is to provide an overview of computer operating systems. Topics covered include: Operating system structures, processes, process synchronization, deadlocks, CPU scheduling, memory management, file systems, Threads and disk management.

**Course Objectives:**

Students will be able to:

1. Explain the fundamental components of the operating system and process concepts
2. Understand the principles of Process management and Deadlocks.
3. Analyze various CPU scheduling algorithms
4. Study the importance of memory management
5. To learn I/O management and File systems

**UNIT I INTRODUCTION**

**9 hours**

Introduction to operating systems -Types of operating systems-Operating System Structures- Operating System Operations- Operating Systems Services-System Calls- System Programs - Process Concept-Process Scheduling-Operations on Processes – Cooperating Processes – Inter-process Communication.

**UNIT II PROCESS MANAGEMENT**

**9 hours**

System Model – Deadlock Characterization – Methods for handling Deadlocks -Deadlock Prevention – Deadlock avoidance – Deadlock detection – Recovery from Deadlocks - Storage Management – Swapping – Contiguous Memory allocation – Paging – Segmentation – Segmentation with Paging

**UNIT III CPU SCHEDULING**

**9 hours**

Threads – Overview – Threading issues - CPU Scheduling – Basic Concepts – Scheduling Criteria – Scheduling Algorithms – Multiple-Processor Scheduling – Real Time Scheduling - The Critical-Section Problem – Synchronization Hardware – Semaphores – Classic problems of Synchronization – Critical regions – Monitors.

#### **UNIT IV MEMORY MANAGEMENT**

**9 hours**

Virtual Memory – Demand Paging – Process creation – Page Replacement – Allocation of frames – Thrashing - File Concept – Access Methods – Directory Structure – File System Mounting – File Sharing – Protection

#### **UNIT V FILE SYSTEM**

**9 hours**

File System Structure – File System Implementation – Directory Implementation – Allocation Methods – Free-space Management. Kernel I/O Subsystems - Disk Structure – Disk Scheduling – Disk Management – Swap-Space Management - Case Study: The Linux System & Windows

#### **Course Outcomes:**

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

1. Explain the Operating Systems basics and process concepts.
2. Analyze various process synchronization approaches
3. Elaborate various CPU scheduling algorithms
4. Classify the various memory management Concepts
5. Implement the concept of file systems.

#### **Text Book(s)**

1. Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, “Operating System Concepts”, 9th Edition, John Wiley and Sons Inc., 2012.
2. Harvey M. Deitel, “Operating Systems”, 2 nd edition, Pearson Education, 2002.

#### **Reference Books**

1. William Stallings, “Operating Systems – Internals and Design Principles”, 7th Edition, Prentice Hall, 2011.
2. Andrew S. Tanenbaum, “Modern Operating Systems”, Second Edition, Addison Wesley, 2001.

**Mode of Evaluation:** Assignments, Mid Term Tests, End Semester Examination.

**MCA I Year I Semester**

**22MCAP105 COMPUTER NETWORKS**

**L T P C**  
**4 0 0 4**

**Pre-requisite**        **NIL**

**Course Description:**

This course provides an introduction to computer networks, with a special focus on the Internet architecture and protocols. Topics include layered network architectures, addressing, naming, forwarding, routing, communication reliability, the client-server model, web and email protocols.

**Course Objectives:**

Students will be able to

1. Understand networking concepts and basic communication model.
2. Understand network architectures and components required for data communication.
3. Analyse the function and design strategy of physical, data link, network layer and transport layer.
4. Acquire basic knowledge of various application protocol for internet security issues and services
5. Understand the working principles of various application protocols and fundamentals of security issues and services available

**UNIT I        NETWORK FUNDAMENTALS**

**9 hours**

Uses of Networks – Categories of Networks -Communication model –Data transmission concepts and terminology – Protocol architecture – Protocols – OSI – TCP/IP – LAN Topology - Transmission media .

**UNIT II        DATA LINK LAYER**

**9 hours**

Data link control - Flow Control – Error Detection and Error Correction - MAC – Ethernet, Token ring, Wireless LAN MAC – Blue Tooth – WiFi-WiMaX.

**UNIT III        NETWORK LAYER**

**9 hours**

Network layer – Switching concepts – Circuit switching – Packet switching –IP — Datagrams — IP addresses- IPV4 and IPV6 – ICMP – Routing Protocols – Distance Vector – Link State-Flooding.

#### **UNIT IV    TRANSPORT LAYER**

**9 hours**

Transport layer –service –Connection establishment – Flow control – Transmission control protocol – Congestion control and avoidance – User datagram protocol. -Transport for Real Time Applications (RTP).

#### **UNIT V    APPLICATIONS AND SECURITY**

**9 hours**

Applications - DNS- SMTP – WWW –SNMP- Security –threats and services - DES- RSA.

#### **Course Outcomes:**

After successful completion of the course, students will be able to

1. Trace the flow of information from one node to another node in the network
2. Identify the components required to build different types of networks
3. Understand the functionalities needed for data communication into layers
4. Choose the required functionality at each layer for given application
5. Understand the working principles of various application protocols and fundamentals of security issues and services available

#### **Text Book(s)**

1. Andrew S.Tannenbaum David J. Wetherall, “Computer Networks” Fifth Edition , Pearson Education 2011
2. William Stallings, —Data and Computer Communications, Tenth Edition, Pearson Education, 2013.

#### **Reference Books**

1. Douglas E. Comer, —Internetworking with TCP/IP (Volume I) Principles, Protocols and Architecture, Sixth Edition, Pearson Education, 2013.
2. Forouzan, “ Data Communication and Networking”, Fifth Edition , TMH 2012.

**Mode of Evaluation:** Assignments, Mid Term Tests, End Semester Examination.

## MCA I Year I Semester

### 22ENGP201 CORPORATE COMMUNICATIONS LABORATORY

<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
<b>0</b>	<b>0</b>	<b>4</b>	<b>2</b>

**Course Prerequisite:** Nil

**Course Description:** English is practical and it is a must for any institution to provide students with opportunities to indulge in actively applying their language skills. Thus the Communication Skills Lab facilitates students with adequate opportunities to put their communication skills in use. It also accommodates peer learning by engaging students in various interactive sessions. This lab will be accompanied by a practical lab component.

**Course Objectives:** This course enables the students to –

1. Focus on their interactive skills
2. Develop their communicative competency
3. Fortify their employability skills
4. Empower their confidence and overcome their shyness
5. Become effective in their overall performance in the industry

#### **UNIT – I LISTENING AND SPEAKING SKILLS 9 hours**

Listening/watching interviews, conversations, documentaries, etc.; Listening to lectures, discussions from TV/Radio/Podcast. Conversational skills (Formal and Informal); Group Discussion; Making effective Power Point presentations;

#### **UNIT – II READING AND WRITING SKILLS 8 hours**

Reading different genres of texts including newspapers Magazines: creative writing; Writing job applications and resume; Emails; Letters; Memorandum; Reports; Writing abstracts and summaries; Interpreting visual texts.

#### **UNIT – III ACCLIMATIZING STUDENTS TO OTHER EXAMS 5 hours**

Test of English as a Foreign Language (TOEFL); International English language Testing System (IELTS); Civil Service Examinations; Verbal--ability.

#### **UNIT – IV INTERVIEW SKILLS 8 hours**

Different types of interviews: Answering questions and offering information; Mock interviews; Body Language; Articulation of sounds; Word Stress, Sentence Stress and Intonation.

**Course Outcomes:** At the end of the course, learners will be able to:

1. Read articles from magazines and newspapers
2. Participate effectively in informal conversations
3. Introduce themselves and their friends and express opinions in English
4. Comprehend conversations and short talks delivered in English
5. Write short essays of a general kind, draft Reports and personal letters and emails in English.

***Suggested Reading/Textbook:***

1. Sanjay Kumar and PushpLata; *Communication Skills*; Oxford University Press, 2012.
2. Sabina Pillai and Agna Fernandez; *Soft Skills and Employability Skills*; Cambridge University Press, 2018.
3. S.P. Dhanavel; *English and Communication Skills for Students of Science and Engineering*; Orient Blackswan, 2009.
4. M. Ashraf Rizvi; *Effective Technical Communication*; Tata Mc Graw Hill Co. Ltd, 2005.

***Reference:***

1. Dr.M.Adithan; *Study Skills for Professional Students in Higher Education*; S.Chand& Co. Pvt., 2014.
2. Guy Brook Hart & Vanessa Jakeman; *Complete IELTS*; Cambridge University Press, 2014.
3. Vanessa Jakeman& Clare Mcdowell; *Action Plan for IELTS*; Cambridge University Press, 2006.
4. Guy Brook Hart; *Instant IELTS*; Cambridge University Press, 2004.
5. S.P.Bakshi&Richa Sharma; *Descriptive General English*; Arihant Publications, 2012.
6. Charles Browne, Brent Culligan 7 Joseph Phillips; *In Focus (level 2)*; Cambridge University Press.
7. Steven Gershon; *Present Yourself 2* (second edition); Cambridge University Press.
8. Leo Jones; *Let's Talk 3*(second edition); Cambridge University Press.
9. Nutall J. C.; *Reading Comprehension*; Orient Blackswan.
10. [www.cambridgeenglish.org/in/](http://www.cambridgeenglish.org/in/)
11. <https://learnenglish.britishcouncil.org/en/english-grammar>
12. <https://www.rong-chang.com/>

**Mode of Evaluation:** Continuous Internal Evaluation, Practical Examination.

## MCA I Year I Semester

### 22MCAP201 PYTHON PROGRAMMING LABORATORY

	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
<b>Course Prerequisite:</b> Any Programming Language	<b>0</b>	<b>0</b>	<b>3</b>	<b>1.5</b>

#### Course Description:

This course provides the students how to write programs in python language to perform different tasks in major concepts such as functions, collections, comprehensions, files, exception handling and OOPS.

#### Course Objectives:

Student will be able to

1. Write algorithms and draw flowcharts using raptor tool for different computational problems
2. Design Python programs using looping, decision making and user defined functions.
3. Develop Python programs using collections, comprehensions, files, exceptions and OOPS

#### LIST OF EXPERIMENTS

##### SIMPLE PROGRAMS USING IF CONSTRUCTS AND CONTROL STRUCTURES

1. Draw flow charts for the following problems using Raptor Tool

- a. To exchange/swap the values of two variables
- b. To find factorial of a given number
- c. To find the sum of set of numbers
- d. To generate Fibonacci sequence
- e. To reverse the digits of a number

2. Develop Python scripts for the problems given in Ex.1

3. Develop Python scripts for the following

- a. To count and display the total number of prime numbers in given two ranges
- b. To calculate the sum of digits of given N digit number
- c. To find the largest of given three numbers

4. Develop Python scripts for the following

- a. To generate Armstrong numbers
- b. To check the given number for perfect or not
- c. To find the factors of the given number

5. Develop Python scripts for the following

- a. To convert the given Centigrade/Celsius temperature into Fahrenheit
- b. To generate multiplication table for the given number with terms
- c. To convert the given decimal number into binary

USING LAMBDA AND UDF

6. Develop Python scripts for the following using Lambda function

- a. To create a lambda function that multiplies argument x with argument y
- b. To filter, odd and even numbers
- c. To square and cube every number in a given list
- d. To extract year, month, date and time
- e. To find palindromes in a given list of strings

7. Develop Python scripts for the following using UDF

- a. To check the given number for Prime or not
- b. To check the given number for odd or even
- c. To calculate the length of given string
- d. To count vowels in given string
- e. To calculate the sum of all items in the given list

USING COLLECTIONS

8. Develop Python scripts for the following without using String built-in functions

- a. To calculate the length of the given string
- b. To reverse the given string
- c. To count and display the Vowels in the given string
- d. To remove spaces from the given string
- e. To convert the given string from lowercase to uppercase

9. Develop Python scripts to implement slicing and indexing operations on the following



- a. String
- b. List
- c. Tuple
- d. Set
- e. Dictionary

10. Develop Python scripts for the following without using List built-in functions

- a. To sum all the items in given list
- b. To count the total number of items in given list
- c. To check the given list is empty or not
- d. To get the largest number from a given list
- e. To shuffle and print the given list

11. Develop Python scripts to implement the following Set operations

- a. Union function
- b. Difference function
- c. Intersection function
- d. All and any functions
- e. Enumerate function

12. Develop Python scripts to do the following in Tuple

- a. To demonstrate packing and unpacking
- b. To check for membership with in and not in operators
- c. Slicing operations
- d. To concatenate more than one tuple into a single tuple
- e. Indexing operations

13. Develop Python scripts for the following in Dictionary

- a. To create a dictionary and add items
- b. To modify existing values using keys
- c. Update function
- d. To print values only

e. To print keys only

14. Develop Python scripts for the following in Array

a. To sort the given array elements

b. To add given two matrix

c. To transpose the given matrix

USING COMPREHENSIONS AND FILES

15. Develop Python scripts for the following in Comprehensions

a. To print a list with if

b. To print a list with if-else

c. To print a list with nested comprehensions

d. To print Dictionary items with a condition using zip function

e. To print Set items

16. Develop Python scripts for the following in Files

a. To create a file with given data

b. To print an existing file data

c. To print student marks with percentage

USING EXCEPTION HANDLING AND OOPS

17. Develop Python scripts for the following in Exceptions

a. To check zero division error

b. To check invalid index reference in str/list/...

c. To handle multiple exceptions

d. To raise an exception

e. To implement user defined exception

18. Develop Python scripts for the following in OOPS

a. To implement a class with attributes and methods

b. To implement a class with constructor

c. To implement single inheritance

d. To implement method overriding

e. To implement encapsulation with all access controls

**Course Outcomes:**

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

1. Draw flow charts for different tasks using raptor tool
2. Write python basic programs using conditional, looping structures and functions
3. Write python programs for collections, comprehensions, files, exceptions and OOPS

**Text Books:**

- 1) Python Programming – An Introduction to computer science, John Zelle, JimLeisy
- 2) Programming and Problem Solving with Python by Ashok Namdev Kamthane and Amit Ashok Kamthane, McGraw Hill Education; First edition (1 November 2017)

**Reference Books:**

- 1) Programming Python, Mark Lutz, O'Reilly, 3<sup>rd</sup> Edition, 2006
- 2) Core Python Programming, Wesley J Chun, PH, 2<sup>nd</sup> Edition
- 3) Python Programming: A Compatible Guide for Beginners to Master and Become an Expert in python programming Language, Brain Draper, CreateSpace Independent Publishing Platform, 2016

**Mode of Evaluation:** Internal Practical Examination, External End Semester Practical Exam

## MCA I Year I Semester

### 22MCAP202 DATABASE MANAGEMENT SYSTEMS LABORATORY

L	T	P	C
0	0	3	1.5

**Course Prerequisite:** None

#### Course Description:

This course provides an executive understanding and approach to the technical subject of database management system.

#### Course Objectives:

Students will be able to:

1. Explain DDL, DML, TCL and DCL Commands
2. Explain date functions Character functions, String functions and Aggregate functions
3. Explain PL/SQL functions, Procedure functions, Triggers and exceptions.

#### List of Experiments:

1. Implement DDL Commands. (use constraints while creating tables).
2. Implement DML commands by using various examples.
3. Implement Unique, NULL, NOT NULL, Primary key, Foreign key, Check constraints.
4. Implement DCL commands, NVL(), Coalesce() functions.
5. Queries (along with sub Queries) using ANY, ALL, IN, EXISTS, NOT EXISTS, UNION, INTERSET.
6. Queries using Aggregate functions (COUNT, SUM, AVG, MAX and MIN), GROUP BY, HAVING.
7. Queries using Conversion functions (to\_char, to\_number and to\_date), string functions (Concatenation, lpad, rpad, ltrim, rtrim, lower, upper, initcap, length, substr and instr)
8. Queries using date functions (Sysdate, next\_day, add\_months, last\_day, months\_between).
9. Implement all types of joins.
10. Calculate Employee Net salary and Gross salary for the following data using cursors. If

Basic salary  $\leq$  8000 then DA is 8.18% of basic,

HRA is 20% of basic PF is 20.18% of basic PT = 60Rs

If Basic salary  $>$  8000 and  $\leq$  14000 then DA is 32% of basic,

HRA is 7% of basic PF = 600Rs

PT = 80Rs

If Basic salary  $>$ 14000 and  $\leq$  16000 then

DA is 30% of basic

HRA is 6% of basic

PF = 600

PT =160

11. Implement functions, procedures and triggers in PL/SQL
12. Implement User defined and System defined exceptions.

**Mini Project:**

Each student has to implement any one of the following system.

- I. Employee management system
- II. Student Result information system
- III. Bank Management system
- IV. Library information system
- V. Hotel management system
- VI. Railway reservation system

Note: Each system contains minimum of 6 tables. Each table contains minimum 10 rows.

**Course Outcomes:**

At the end of the course the student will be able to:

1. Execute DDL, DML and DCL commands.
2. Implement PL/SQL procedures, functions, cursors, triggers and exceptions
3. Design Relational model for the given system.

**Text Books:**

1. Benjamin Rosenzweig, Elena Silvestrova, ORACLE PL/SQL by example, Pearson Education
2. Dr.P.S. Deshpande, SQL & PL/SQL for Oracle 10g, Black Book.

**Reference Books:**

1. Rick F.Vander Lans, Introduction to SQL, Pearson Education.
2. Steven Feuerstein, Oracle PL/SQL Programming, SPD.
3. N.Gehani, The Database Book, Universities Press.
4. Shah, Database Systems using Oracle: A Simplified Guide to SQL and PL/SQL, PHI.

**Mode of Evaluation:** Continuous Cumulative evaluation of the lab experiments, Record, Viva voce and External Lab Examination.

## SKILL-ORIENTED COURSE

MCA I Year I Semester

22MCAP601 PROGRAMMING WITH C++

L T P C

1 0 2 2

**Pre-requisite:** Basic Programming Knowledge

### Course Description:

This course contains the basics of object oriented programming fundamentals through C++. It contains expressions, control flow, functions, arrays, pointers, managing I/O and object oriented programming features.

### Course Objectives:

Students will be able to:

1. Understand the basic concepts of object oriented programming.
2. Understand the basic concepts of Functions.
3. Explain Class, Object, Constructor & Destructor and pointers.
4. Demonstrate inheritance, operator overloading and dynamic memory allocation concepts and Describe polymorphism and generic constructs
5. Explain streams, File I/O and exception handling in C++.

### UNIT I

(6)

**Introduction to OOP:** Object oriented programming, Basic concepts of OOP, Benefits, and applications of OOPS, User defined types

**C++ Basics:** Structure of a C++ program, Token, Data types, Declaration of variables, Expressions, Operators

- 1) Develop a C++ program to find all roots of a quadratic equation  $ax^2+bx+c=0$ .
- 2) Develop a C++ program find the largest among three different numbers entered by user.
- 3) Develop a C++ program to find the Fibonacci series till the limit entered by the user.
- 4) Write a C++ program to print the following by reading number of rows to be printed from user

```
*
* *
* * *
* * * *
* * * * *
```

- 5) Develop a C++ program to find the sum of digits of a number

## UNIT II

(6)

**C++ Functions:** Simple functions, 'Call and Return' by reference, Inline functions, Function overloading, Function Over riding, Default arguments, pointers and functions

- 1) Write a C++ program to find the value of a number raised to its power that demonstrates a function using call by value.
- 2) Write a C++ program to implement recursion in finding the factorial.
- 3) Write a C++ program write a program to find the multiplication values and the cubic values using inline function.
- 4) Write a C++ program to calculate the volume of cube, cylinder and rectangular box using function overloading.

## UNIT III

(6)

**Classes and Objects:** Class, Member functions, Access Specifiers, Static data members and member functions, Arrays of objects, Returning objects, Friend class and friend functions.

**Constructors and Destructors:** Constructors, Default Constructors, Copy Constructor, Constructor overloading, Destructors, Use of "this" pointer, dynamic memory allocation

**Pointers in C++:** Introduction-Pointers and arrays, Pointers to objects, this pointer, Pointers to functions, Call by pointer.

- 1) Write a C++ program to display student's information using Class Declarations, Definition, and Accessing Class Members by objects.
- 2) Define a class to represent a Bank account. Include the following members.

**Data members:** Name of the depositor; Account number, Type of account, Balance amount in the account, Rate of interest (static data).

Provide a default constructor; a parameterized constructor and a copy constructor to this class Also provide **Member Functions:** - 1. To deposit amount. 2. To withdraw amount after checking for minimum balance. 3. To display all the details of an account holder. 4. Display rate of interest (a static function) Illustrate all the constructors as well as all the methods by defining objects and destroy the object using destructor.

- 3) Develop a C++ program to find the area of a rectangle by converting the member of a class square which is a friend class of rectangle. Declare Rectangle as a friend of Square so that Rectangle member functions could have access to the private member of square.
- 4) Write a C++ program to implement flight class with data member as flight number, source, destination and fare. Write a member function to display the flight information using this pointer.
  - a) Write C++ Program to store GPA (Grade Point Average) of n number of students and display it using new and delete operator

## UNIT IV

(6)

**Inheritance:** Single inheritance, Multi-level inheritance, Multiple inheritance, Hierarchical inheritance, Hybrid inheritance. Operator Overloading.

**Virtual Functions and Run Time Polymorphism:** Overriding, Static and Dynamic binding, Virtual functions, Pure virtual functions, Abstract classes.

- 1) Write a C++ program to accept details of n instructors and display them. Create base Class: Person (id, name) and Derive Two Sub Classes: Teaching (Subject, Name), Nonteaching(dept.) from class Person and derive new class Instructor from two sub classes Teaching and Nonteaching.
- 2) Write a C++ program to explain virtual function (polymorphism) by creating a base class c\_polygon which has virtual function area (). Two classes c\_rectangle and c\_traingle derived from c\_polygon and they have area () to calculate and return the area of rectangle and triangle respectively
- 3) Write a C++ program to count the number of persons inside a bank, by increasing count whenever a person enters a bank, using an increment (++) operator overloading function, and decrease the count whenever a person leaves the bank using a decrement (--) operator overloading function inside a class.

## UNIT V

(6)

**Templates and Streams in C++:** Generic Functions - Generic Classes - Console stream classes, Formatted and unformatted console I/O operations, Manipulators; File stream classes, File I/O.

**Exception Handling:** Principles of Exception handling, Exception handling mechanism, Multiplecatch, Nested try.

- 1) Write a C++ program create a template T for a class named pair having two data members of type T which are inputted by a constructor and a member function get-max () return the greatest of two numbers to main.
- 2) Write a C++ program to perform the deletion of white spaces such as horizontal tab, vertical tab, space, line feed, new line and carriage return from a text file and store the contents of the file without the white spaces on another file.
- 3) Write a C++ program to read the class object of student info such as name, age, sex, height and weight from the keyboard and to store them on a specified file using read () and write () functions. Again the same file is opened for reading and displaying the contents of the file on the screen



**Course Outcomes:**

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

1. Explain the concepts of object-oriented programming and C++ basics.
2. Explain the concepts of object-oriented programming functions
3. Implement class and object Constructor & Destructor and pointers .
4. Demonstrate inheritance, operator overloading and dynamic Apply inheritance and operator overloading concepts and Develop the programs using polymorphism, function and class templates.
5. Implement streams, File I/O and exception handling in C++.

**Text Book(s)**

1. E. Balagurusamy, “Object Oriented Programming with C++”, Tata McGraw-Hill, 7th Edition, 2018.
2. Schildt Herbert, The Complete Reference C++, Tata McGraw Hill, 4th Edition, 2009.

**Reference Books**

1. Deitel and Deitel, C++ How to program, Pearson Education Asia, 9<sup>th</sup> Edition, 2014
2. Yeshavant Kanetkar, Let Us C++, Second Edition, BPB Publications, 2003.
3. Lafore, Robert, Object Oriented Programming in Turbo C++, Galgotia Publications Pvt. Ltd, 2000.
4. Gaddis Tony, Starting Out with C++, dreamtech Press, 3rd Edition, 2002.
5. Sotter A Nicholas and Kleper J Scott, Professional C++, Wiley Publishing Inc.
6. Walter Savitch, Problem solving with C++, Pearson Education, 6th Edition 2007.

**Mode of Evaluation:** Assignments, Internal Mid Examination, External End Lab Examination.

# **MCA I Year II Semester**

## MCA I Year II Semester

### 22MCAP106 FULL STACK WEB DEVELOPMENT

**L T P C**  
**4 0 0 4**

**Pre-requisite** Computer Networks Basics

#### **Course Description:**

This course introduces web designing and development methodologies using the front end development languages such as HTML, CSS, Javascript and Angular JS. Also, this course gives the idea of web development process using NODE JS.

#### **Course Objectives:**

Student will be able to:

1. Get familiar with web design using HTML and CSS
2. Understand the concepts on Bootstrap framework and JavaScript
3. Identify the working methodologies on Angular JS concepts
4. Get familiarity with Node JS techniques
5. Gain the knowledge about database connectivity for web application development

#### **UNIT I HTML & CSS**

**9 hour**

HTML Basics HTML Elements --HTML Attributes - HTML Formatting - HTML Links and Images HTML Lists, HTML tables – Frames and Framesets - Blocks, Classes - HTML Layout and Forms. CSS: Introduction CSS3 -□CSS3Colours - Backgrounds, Borders, Padding, Height/Width - CSS3 Gradients, Shadows - CSS3 Text, Fonts - CSS3 2D &3D Transforms - CSS Links – CSS Lists & Tables - CSS Box Model, Outline, Display, Max-width, Position - CSS Float, Inline-block – CSS Align – CSS Pseudo-class, Pseudo-element - CSS Navigation, Dropdowns, Tooltips, Images – CSS Selectors - CSS Forms, Buttons - CSS3 Multiple Columns - CSS3 User Interface: Box Sizing, Filters – Menu creation -Responsive CSS

#### **UNIT II BOOTSTRAP AND JAVASCRIPT**

**9 hours**

Introduction to Bootstrap - Bootstrap Basics - Bootstrap Grids - Bootstrap Themes – Bootstrap with CSS Introduction to JavaScript - Java Script Language Basics -Scope - JavaScript Events - Strings JavaScript Math - Arrays – Boolean – Comparisons - JavaScript Loops & Decisions - JavaScript objects and Method - JavaScript Errors - Debugging - JavaScript Functions - JavaScript Forms - JavaScript DOM

#### **UNIT III ANGULAR JS**

**9 hours**

Introduction – Features of Angular JS – Angular Expressions – Directives – Controllers – Modules – Forms – Dependency Injection and Services – Angular JS Animations

**UNIT IV    NODE JS****9 hours**

Stream Data Model and Architecture - Data Stream Management Systems – Data Stream Mining and Examples of Data Stream Applications - Mining Time Series Data –Stream Queries – Issues in Data Stream Query Processing – Sampling in Data Streams –Filtering Streams – Counting Distinct Elements in a Stream. Counting Ones in a Window

**UNIT V    DATABASE CONNECTIVITY WITH NODE JS****9 hours**

Overview of Database Queries – Connecting String – Configuring Node JS – Working with select Command – Updating Records – Deleting Records –MERN STACK - Project Development Using Node JS

**Course Outcomes:**

After completion of the course, students will be able to

1. Write HTML and CSS scripts to design web pages.
2. Implement JavaScript methodologies while designing web pages
3. Design web pages using Angular JS techniques
4. Implement Node JS concepts in web development
5. Develop web sites using Node JS with database connectivity.

**Text Book(s)**

1. HTML & CSS: The Complete Reference, Fifth Edition, Thomas Powell, McGrawHill
2. Complete Bootstrap: Responsive Web Development with Bootstrap 4, Matt Lambert,Bass Johnson, David Cochran, Ian Whitley, PACKT publishing
3. Javascript for Dummies, Emily Vander Veer, Wiley Publishing, 4<sup>th</sup> Edition
4. Learning Angular JS, Ken Williamson, O.Reilly, 2015
5. Beginning Node JS, Basarath Ali Syed, Apress, 2014

**Reference Books**

1. Complete Bootstrap: Responsive Web Development with Bootstrap 4, Matt Lambert,Bass Johnson, David Cochran, Ian Whitley, PACKT publishing
2. Mastering Javascript, VedAntani, PACKT publishing, 2016
3. Node JS Web Development, David Herron, PACKT publishing, 2016

**Mode of Evaluation:** Assignments, Mid Term Tests, End Semester Examination.

**MCA I Year II Semester**

**22MCAP107 DATA STRUCTURES AND ALGORITHMS**

**L T P C**  
**3 1 0 4**

**Pre-requisite**        **NIL**

**Course Description:**

The purpose of this course is to impart knowledge on various data structures and design and analyze the algorithm concepts.

**Course Objectives:**

1. Demonstrate the algorithms and basic concepts of data structures.
2. Analyze and implement linear data structures with linked list.
3. Explore and implement non- linear data structures with trees and graphs.
4. Explain different types of hashing and sorting techniques.
5. Analyze different types of Problem-Solving Methods.

**UNIT I        INTRODUCTION**

**9 hours**

Introduction of Algorithms - Analyzing Algorithms - Arrays: Sparse Matrices Representation of Arrays - Stacks and Queues: Fundamentals - Evaluation of Expression - Infix to Postfix Conversion - Multiple Stacks and Queues - Perform Analyze the Algorithms.

**UNIT II        LINEAR DATA STRUCTURES**

**9 hours**

Linked List: Singly Linked List – Doubly Linked List – Circular Linked List - Linked Stacks and Queues - Polynomial Addition - Sparse Matrices - Dynamic Storage Management - Garbage Collection and Compaction.

**UNIT III        NON - LINEAR DATA STRUCTURES**

**9 hours**

Trees: Basic Terminology - Binary Trees - Binary Tree Representations - Threaded Binary Trees - Binary Tree - Representation of Trees.  
Graphs: Terminology and Representations – Traversals - Connected Components and Spanning Trees.

**UNIT IV        HASHING AND SORTING TECHNIQUES**

**9 hours**

Introduction to Hashing, Deleting from Hash Table, Efficiency of Rehash Methods, Hash Table Reordering, Resolving collision by Open Addressing.  
Sorting Techniques: Bubble sort – Insertion sort - Quick sort – merge sort – heap sort - Divide and Conquer method.

## **UNIT V      PROBLEM-SOLVING METHODS**

**9 hours**

Dynamic Programming Methods: All pairs shortest paths, Travelling salesman problem; Backtracking- N-queen problem, Graph coloring; Branch and Bound-15-puzzle.

### **Course Outcomes:**

After successful completion of the course, students will be able to

1. Analyze the complexity of algorithms.
2. Applying Linear data structure concepts and algorithms.
3. Applying Non-linear data structures concepts and algorithms.
4. Compare various hashing and sorting techniques.
5. Implement various types of problem-solving approaches.

### **Text Book(s)**

1. Fundamentals of Data structures, E. Horowitz, S. Sahni and Susan Anderson-Freed, Universities Press.
2. EllisHorowitz, SartajShani "Data and File Structures', GalgotiaPublication, Second Edition,2013.

### **Reference Books**

1. Benjamin Baka , Python Data Structures and Algorithms, Packt Publishing 2016.
2. Rance D. Necaise "Data Structures and Algorithms Using Python", John Wiley & Sons, Inc.

**Mode of Evaluation:** Assignments, Mid Term Tests, End Semester Examination.

**MCA I Year II Semester**

**22MCAP108 SOFTWARE ENGINEERING**

**L T P C**

**4 0 0 4**

**Pre-requisite**            **NIL**

**Course Description:**

Software engineering is the branch of computer science that creates practical, cost-effective solutions to computing and information processing problems, preferentially by applying scientific knowledge, developing software systems in the service of mankind. This course covers the fundamentals of software engineering, including understanding system requirements, finding appropriate engineering compromises, effective methods of design, coding, and testing, team software development.

**Course Objectives:**

The course is designed to meet the objectives of:

1. The aim of the course is to provide an understanding of the working knowledge of the techniques for estimation, design, testing and quality management of large software development projects.
2. Topics include process models, software requirements, software design, software testing,
3. software process/product metrics, risk management, quality management and UML diagrams.

**UNIT I            INTRODUCTION TO SOFTWARE ENGINEERING AND            9 hours**  
**PROCESS MODELS**

Introduction to Software Engineering: The evolving role of software, changing nature of software, software myths. A Generic view of process: Software engineering- a layered technology, a process framework, the capability maturity model integration (CMMI), process patterns, process assessment, personal and team process models. Process models: The waterfall model, incremental process models, evolutionary process models, the unified process.

**UNIT II            SOFTWARE REQUIREMENTS & MODELS            9 hours**

Functional and non-functional requirements, user requirements, system requirements, interface specification, the software requirements document. Requirements engineering process: Feasibility studies, requirements elicitation and analysis, requirements validation, requirements management. System models: Context models, behavioural models, data models, object models, structured methods.

**UNIT III            DESIGN ENGINEERING            9 hours**

Design Engineering: Design process and design quality, design concepts, the design model. Creating an architectural design: software architecture, data design, architectural styles and patterns, architectural design, conceptual model of UML, basic structural modelling, class diagrams, sequence diagrams, collaboration diagrams, use case diagrams, component diagrams.

#### **UNIT IV TESTING STRATEGIES**

**9 hours**

Testing Strategies: A strategic approach to software testing, test strategies for conventional software, black-box and white-box testing, validation testing, system testing, the art of debugging. Product metrics: Software quality, metrics for analysis model, metrics for design model, metrics for source code, metrics for testing, metrics for maintenance.

#### **UNIT V SOFTWARE RELIABILITY AND QUALITY MANAGEMENT**

**9 hours**

Software Reliability And Quality Management: Software Reliability, Statistical Testing, Software Quality, Software Quality Management System, ISO 9000, SEI Capability Maturity Model. Software Maintenance: Characteristics of Software Maintenance. Software Reuse: what can be Reused? Why almost No Reuse So Far?, Basic Issues in Reuse Approach, Reuse at Organisation Level.

#### **Course Outcomes:**

After successful completion of the course, students will be able to

1. Understand the role of software and various process models (L2).
2. Analyze the software project from requirement gathering to implementation (L4).
3. Apply various UML diagrams to the software project (L3).
4. Assess the testing strategies and product metrics (L5).
5. Understand knowledge about software reliability and quality management (L2).

#### **Text Book(s)**

1. Software Engineering: A Practitioner's Approach, Roger S. Pressman, 10th ed, Mc Graw Hill
2. Rajib Mall, "Fundamentals of Software Engineering", 5th Edition, PHI, 2018.
3. The unified modeling language user guide Grady Booch, James Rumbaugh, Ivar Jacobson, Pearson Education.

#### **Reference Books**

1. Software Engineering, 8/e, Sommerville, Pearson
2. Software Engineering principles and practice, W S Jawadekar, TMH
3. Craig Larman, Applying UML and Patterns: An Introduction to object- Oriented Analysis and Design and iterative development,3rd Edition, Pearson,2005.

**Mode of Evaluation:** Assignments, Mid Term Tests, End Semester Examination.



**MCA I Year II Semester**

**22MCAP109 CRYPTOGRAPHY AND NETWORK SECURITY**

**L T P C**

**3 1 0 4**

**Pre-requisite: Computer Networks**

**Course Description:**

Network Security introduces techniques for protecting information and network components against attacks. It highlights the core cryptographic mechanisms and their implementation procedure to provide security for the data. Investigates various networking security standards and methods for enforcing and enhancing those standards. It also covers the electronic mailing system and supporting protocols.

**Course Objectives:**

Students will be able to

1. Gain knowledge about symmetric key encryption techniques
2. Study of asymmetric key cryptographic algorithms and encryption methods for security.
3. Understand the message authentication codes and hashing algorithms.
4. Understand the method of public, private key distribution and applications related to authentication.
5. Know about IP security and Transport layer security with supporting protocols

**UNIT I SYMMETRIC CIPHERS**

**9 hours**

Introduction to security attacks, services and mechanisms, Classical Encryption Techniques – Substitution Ciphers and Transposition Ciphers, Stream and Block Ciphers. Modern Block Ciphers: Block cipher principles, Shannon's theory of confusion and diffusion, feistel structure, modular arithmetic, DES, AES.

**UNIT II ASYMMETRIC CIPHERS**

**9 hours**

Prime and Relative Prime numbers, Extended Euclidean Algorithm, Fermat's and Euler's theorem, primality testing, Chinese Remainder Theorem, Discrete Logarithmic Problem, Principle of Public key crypto systems, RSA algorithm, security of RSA.

### **UNIT III MESSAGE AUTHENTICATION**

**9 hours**

Message Authentication Codes: Authentication requirements, authentication functions, message authentication code, hash functions, security of hash functions, Secure hash algorithm (SHA). Digital Signatures: Digital signature standards (DSS).

### **UNIT IV KEY DISTRIBUTION & AUTHENTICATION APPLICATIONS**

**9 hours**

Key Management and distribution: Symmetric key distribution, Diffie-Hellman Key Exchange, Public key distribution, X.509 Certificates, Public key Infrastructure (PKI). Authentication Applications: Kerberos, Electronic mail security: Secure Multipurpose Internet Mail Extensions (S/MIME).

### **UNIT V NETWORK & TRANSPORT LAYER SECURITY**

**9 hours**

IP Security: Architecture, Authentication header, Encapsulating security payloads. Introduction to Secure Socket Layer, Secure electronic transaction (SET). System Security: Introductory idea of Intrusion, Intrusion detection systems, Viruses and related threats.

#### **Course Outcomes:**

After Completion of the course, the student will be able to

1. Acquire knowledge about block ciphering and symmetric ciphering techniques.
2. Implement RSA algorithm
3. Execute Secure Hashing and Digital Signature Algorithms.
4. Implement authentication applications Kerberos and MIME.
5. Apply Knowledge on IPsecurity, Transport layer security Protocols and system security

#### **Text Book(s)**

1. William Stallings, "Cryptography and Network security Principles and Practices", Pearson/PHI, 4th Edition
2. Behrouz A. Frouzan, "Cryptography and Network Security", Tata McGraw Hill, 2nd Edition

#### **Reference Books**

1. Bruce Schneier, "Applied Cryptography". John Wiley & Sons
2. Bernard Menezes, "Network Security and Cryptography", Cengage Learning
3. Atul Kahate, "Cryptography and Network Security", TMH

**Mode of Evaluation:** Assignments, Mid Term Tests, End Semester Examination.

## MCA I Year II Semester

### 22MCAP110 ARTIFICIAL INTELLIGENCE

L	T	P	C
4	0	0	4

**Course Prerequisites:** None

#### **Course Description:**

To understand the importance of AI techniques and its wide range of applications. The course is introduced with basics of AI and ML. Then and applications and techniques of AI in various domains

#### **Course Objectives:**

Students will be able to:

1. Demonstrate the fundamentals of AI and ML
2. Summarize NLP, Chatbots and Speech Recognition
3. Demonstrate the concepts of Image Processing and Computer Vision
4. Discuss reinforcement learning and its applications
5. Introduced to various real time applications that are implemented using AI

#### **UNIT I: INTRODUCTION TO AI & ML**

**9 hours**

AI: What is AI, Applications of AI, Advanced search, Constraint satisfaction problems, Knowledge representation and reasoning, Uncertain and probabilistic reasoning - ML: Introduction, Types of Machine Learning, Perceptron, Neural Networks, Applications of Machine Learning

#### **UNIT II: NLP, CHATBOTS & SPEECH RECOGNITION**

**9 hours**

Natural Language Processing: Introduction, Natural Language Understanding, Components of NLP, Enterprise Applications, how to Use NLP, Challenges of NLP - Chatbots: Introduction, how to Build a Chatbot, Architecture of Chatbot, Challenges of building a Chatbot, Industry case studies - Speech Recognition: Speech Fundamentals, Speech Analysis, Speech Modeling, Speech Recognition, Speech Synthesis, Text to Speech, Virtual Assistants: What is a Virtual Assistant?

#### **UNIT III: IMAGE PROCESSING & COMPUTER VISION**

**9 hours**

Image Processing: Introduction, Image Noise Removal of Noise, Removal of Noise from Images, Colour Enhancement, Segmentation, Edge Detection, Optical Character Recognition, Feature Detection & Recognition, Feature Extraction - Computer Vision: Capabilities of Computer Vision for an Enterprise, how to use Computer Vision, Computer Vision on Mobile Devices, Existing Challenges and Implementation - Agriculture [Crop and Soil Monitoring, Predictive Analytics], Retail and Retail Security [Amazon Go]

## **UNIT IV: REINFORCEMENT LEARNING**

**9 hours**

Reinforcement Learning: Introduction, Game Playing [ Deep Blue in Chess, IBM Watson in Jeopardy, Google's DeepMind in AlphaGo], Agents and Environment, Action-Value Function, Applications: Robotics, Gaming, Diagnostic systems, Virtual Assistants

## **UNIT V: SMART APPLICATIONS**

**9 hours**

Smart Manufacturing, Smart Agriculture, Smart Grids, Smart Transportation and Autonomous Vehicles, Smart Homes, Smart Cities.

### **Course Outcomes:**

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

1. Demonstrate the basic concepts of AI , ML and its applications
2. Discuss briefly about NLP, Chatbots and Speech Recognition
3. Illustrate the concepts of Image Processing and Computer Vision
4. Demonstrate reinforcement learning and its applications
5. Design smart applications for various domains

### **Textbooks:**

1. Tom Markiewicz& Josh Zheng,Getting started with Artificial Intelligence, Published by O'Reilly Media,2017
2. Stuart J. Russell and Peter Norvig,Artificial Intelligence A Modern Approach
3. Joseph Howse, Prateek Joshi, Michael Beyeler - Opencv\_ Computer Vision Projects with Python-Packt Publishing (2016)

### **Reference Books:**

1. A classical approach to Artificial Intelligence, Munesh Chandra Trivedi, Khanna Publications
2. Richard Szeliski, Computer Vision: Algorithms and Applications, Springer 2010
3. Artificial Intelligence and Machine Learning, Chandra S.S. & H.S. Anand, PHI Publications
4. Machine Learning, Rajiv Chopra, Khanna Publishing House
5. Build an AI Assistant with Wolfram Alpha and Wikipedia in Python.  
<https://medium.com/@salisuwy/build-an-ai-assistant-with-wolfram-alpha-and-wikipedia-in-python-d9bc8ac838fe>

**Mode of evaluation:** Assignments, Internal Mid Examination, External End Examination

## MCA I YEAR II SEM

### 22MCAP203 FULL STACK WEB DEVELOPMENT LABORATORY

L T P C  
0 0 3 1.5

**Pre-requisite** Basic knowledge on Java, DBMS

#### **Course Description:**

Full Stack Web Development course will make students to become master in front-end technology. It provides basic information and experiments to grow to be a Full-Stack web developer. With fast growing technologies, the students can update their knowledge on technologies. This will help the students to learn the complete set of process like designing, development and deployment.

#### **Course Objectives:**

Students will be able to:

1. Acquire knowledge on web designing using front end tools
2. Develop coding using scripting languages
3. Develop applications using AJAX

#### **List of Programs**

1. Write a HTML Program to create a Website for your department using all HTML tags.
2. Write a Java Script program to perform Arithmetic Operations
3. Write a Java Script Program
  - i. To Sort an Array of Strings
  - ii. To print the array values using for loop
4. Develop a Javascript program to print the Even numbers less than the given number using do-while loop
5. Write a Javascript program to
  - i. Check whether the person is eligible for vote or not using if statement
  - ii. Print the Message “Good Morning”, “Good Afternoon” and “Good Evening” based on system timings. (Use If Else)
6. Write Javascript programs to perform all relational operations and show “True” or “False”
7. Write a Javascript program
  - i. To count the number of vowels using functions
  - ii. To perform the arithmetic operations using functions
8. Develop a web page using HTML, CSS and Javascript to display a Calculator. If the button is clicked for Normal Calculator it should display normal calculator. If the button is clicked for Scientific Calculator it should display the scientific calculator
9. Develop a Javascript program to create a validation form
10. Write a Javascript program to demonstrate the Javascript DOM Events
11. Create a form that collects the first name, last name, email, user id, password and confirm password from the user. All the inputs are mandatory and email address entered should be in

correct format. Also, the values entered in the password and confirm password textboxes should be the same. After validating using JavaScript, display proper error messages in red color just next to the textbox where there is an error.

12. Write a Java Script Program to compute the average and grade of the student

13. Design an XML document to store information about a student in an engineering college affiliated to VTU. The information must include USN, Name, and Name of the College, Branch, Year of Joining, and email id. Make up sample data for 3 students. Create a CSS style sheet and use it to display the document

### **Course Outcomes:**

After completion of the course, students will be able to

1. Implement webpages using HTML and CSS
2. Execute different types of Javascript programs
3. Apply the concepts of XML for web programming

### **Text Book(s)**

1. Java Script for Programmers Paul J. Deitel, Deitel & Associates, Inc. Harvey M. Deitel, Deitel & Associates, Inc.
2. **XSLT: Working with XML and HTML** Khun Yee Fung, Addison Wesley, 2000

### **Reference Books**

1. **Web Coding Bible, An Accelerated Course, Chong Lip Phang, 2015**

**Mode of Evaluation:** Continuous Internal Evaluation and End Semester Examination

## MCA I Year II Semester

### 22MCAP204 DATA STRUCTURES AND ALGORITHMS LABORATORY

L T P C

0 0 3 1.5

**Pre-requisite** -NIL-

#### **Course Description:**

This course introduces on the practical part of Data Structures and Algorithms using Python language. This course allows students to implement linear and nonlinear data structures. It gives practical exposure for solving sorting and searching problems. It enables to develop programs using various problem solving methods.

#### **Course Objectives:**

Students will be able to

1. Design Linear and Non-Linear Data structures.
2. Compare different types of searching and sorting techniques.
3. Demonstrate various problem-solving methods.

#### **List of Programs:**

1. Create a Stack and do the following operations.  
a. Push b. Pop c. Peep
2. Create a queue and do the following operations.  
a. Add b. Delete c. Display
3. Write a program to convert from infix to postfix expression.
4. Implement the operations on the following data structures:  
a. singly linked list b. circular linked list c. doubly linked list
5. Implement the operations on Binary Search tree.
6. Write a program on Tree Traversal  
a. Pre order b. In order c. Post order
7. Write a program to find minimum spanning tree using  
a. Prim's method b. Kruskal's method.
8. Write a program to sort elements using the following sorting techniques  
a. Insertion b. Selection c. Bubble.
9. Implement the following Divide and Conquer techniques to sort elements.  
a. Quick sort b. Merge sort
10. Implement single source shortest path problem.
11. Implement all pairs shortest path problem.
12. Implement N-queen's problem using backtracking.

**Course Outcomes:**

After completion of the course the student will be able to

1. Select a suitable data structure for real time applications.
2. Implement Linear and Non-Linear Data structures.
3. Apply various problem-solving techniques for complex problems.

**Text Book(s)**

1. Data Structure and Algorithms in Python by Michael T. Goodrich, and Roberto Tamassia O'reilly and wiley.
2. Data Structures and Algorithm using Python by Brad Miller and David Ranum, Luther College.
3. Problem-Solving with Algorithms and Data Structures Using Python by Bradley N. Mille, Franklin Beedle & Associate Incorporated.

**Reference Books**

1. Data Structures, S. Lipschutz, Schaum's Outlines, TMH.
2. Data Structures and Algorithms ,Alfred Aho,John E. Hop Croft, Ullman, Addison Wesley.
3. Algorithm Design by Jon Kleinberg, Pearson.

**Mode of Evaluation:** Continuous Internal Evaluation and End Semester Examination



## MCA I Year II Semester

### 22MCAP205 SOFTWARE ENGINEERING LABORATORY

L T P C

0 0 3 1.5

**Pre-requisite** NIL

#### **Course Description:**

This course focuses on providing hands-on experience in designing and developing software systems. Specifically, the course studies software modelling tools, and testing tools.

#### **Course Objectives:**

1. To have hands on experience in developing a software project by using various software engineering principles and methods in each of the phases of software development.
2. To practice the various design diagrams through the appropriate tool.
3. To learn to implement various software testing strategies

#### **List of Programs:**

1. Introduction to UML, To Develop Data Dictionary and Use case Diagram
2. To Develop Activity diagram and Class diagram
3. To Develop Sequence diagrams and Collaboration Diagram
4. To add interface to class diagram
5. To Develop Deployment diagram
6. To Prepare test plan and perform validation testing.
7. Develop test cases for unit testing and integration testing
8. Develop test cases for various white box and black box testing techniques.

#### **Course Outcomes:**

After successful completion of the course, students will be able to

1. Design and Implement UML diagrams for real word applications.
2. Implement various testing strategies for software projects.
3. Develop test cases for various testing techniques.

#### **Text Book(s)**

1. Software Engineering? A Practitioner's Approach, Roger S. Pressman, 1996, MGH.

#### **Reference Books**

1. Software Engineering by Ian Sommerville, Pearson Edu, 5th edition, 1999
2. An Integrated Approach to software engineering by Pankaj Jalote , 1991 Narosa.

**Mode of Evaluation:** Continuous Internal Evaluation and End Semester Examination

**Skill Oriented Course – II**  
**MCA I Year II Semester**

**22MCAP602      PROGRAMMING WITH MATLAB**

L	T	P	C
1	0	2	2

**Pre-requisite**      **20MAT101, 20EEE101**

**Course Description:**

This course introduces students to MATLAB programming, and demonstrate its use for scientific computations. The basis of computational techniques is expounded through various coding examples and problems. The practical ways to use MATLAB will be discussed.

**Course Objectives:**

This course enables students to

1. Understand basic MATLAB commands and elementary functions
2. Study and implement mathematical operations and matrices manipulation
3. Understand MATLAB functions and expressions
4. Apply flow control and files in MATLAB
5. Understand Plotting and Simulink blocks in MATLAB

**UNIT I      MATLAB BASICS**

**6 hours**

Introduction, Matlab environment, Matlab as a calculator, Matlab Online, Syntax and Semantics, Help, Data Types-Matrix, string, cell and structure, Variables and Arrays, **Initializing Variables**, Multidimensional Arrays, Sub arrays, Special Values, Displaying Output Data, Data Files, Scalar and Array Operations, Hierarchy of Operations, Built-in MATLAB Functions, Debugging MATLAB Programs

- Swap the values in two variables without using temporary variable. For example, the variable 'x' contains the value '5' and the variable 'y' contains the value '10'. The program should swap the values in the variable's 'x' and 'y'. After the execution of the program the value in the variable 'x' should be '10' and the value in the variable 'y' should be '5'. This should be accomplished without using the temporary variable.
- Write a function which should return either maximum or minimum value of the element in an array.
- Write a code to find whether the given number is even or not.
- Write a function that should sort the elements in the array either in the ascending order or descending order.
- Write a program which should count the number of occurrences of particular element in the array.

**UNIT II      MATRICES AND OPERATORS**

**6 hours**

Introduction, Colon Operator, Accessing Parts of a Matrix, Combining and Transforming Matrices, Arithmetic operations

- Write a program to find the maximum and minimum value of the elements of the matrix

- Write a program to compute the sum of diagonal elements of the given matrix
- Write a program to test whether the given matrix is symmetric or not?

- Obtain the rank of the following matrices (i)  $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$  (ii)  $\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$  and

comment on the result.

- Write a program to check whether the given matrix is invertible or not? {Hint: A matrix is invertible if is not singular. The determinant of the matrix should not be equal to zero}
- Write a program to check the given matrix is orthogonal or not?
- Use the built-in function to compute the eigen value and the eigen vector of the given matrix. From the eigen value is it possible to find whether the given matrix is (i) Positive definite (ii) Positive semidefinite.
- Create a vector 'x' that should contain elements from 1 to 10. Write a code to perform the following operation
  - (i) Add a constant (say 3) to each element of 'x'.
  - (ii) Make all the even indexed elements to zero.
  - (iii) Make all the odd indexed elements to zero.
  - (iv) Generate 'y' which should contain elements in the reverse order of 'x'.
  - (v) Generate 'y' such that it should have first five elements of 'x' and the remaining elements to zero
  - (vi) Add the constant to odd indexed elements of 'x'.
  - (vii) Add the constant to the even indexed elements of 'x'
- Write a program to solve the linear algebraic equation
  - (i)  $5x-3y+2z = 10$
  - (ii)  $-3x+8y+4z = 20$
  - (iii)  $2x+4y-9z = 9$
- Write a program to determine the eigen vector and eigen values of  $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$

### UNIT III FUNCTIONS AND EXPRESSIONS

6 hours

Introduction, Function I/O, Formal Definition of Functions, Sub functions, Scope, Advantages of Functions, Scripts, and Problem-Solving **File Input-Output, Expressions**, write a function which returns the "median" of the array of elements.

- Write a code to print the prime numbers from one to hundred.
- Write a function which accepts the radius of the circle as input and returns the area and perimeter of the circle.
- Write a code which will compute sum of integers ranging from 1 to 100.
- Write a code to compute the "body mass index". The input to the code should be (i) Weight

and (ii) Height of the person. The output of the program should be “body mass index (bmi)”

- Write a program to convert the temperature in degrees to Celsius.
- Write a program to check whether the given string (word) is palindrome or not?
- Write a program to compute the factorial of the given number.
- Find the roots of the polynomial

$$f(x) = 3x^6 + 15x^5 + 10x^3 + 4x$$

- An R-L-C circuit has R = 180 ohms, C = 1/280 farads, L = 20 Henries and an applied voltage E(t) = 10 sin t. Assuming that no charge is present but an initial current of I ampere is flowing at t = 0 when the voltage is first applied, find q and  $i = \frac{dq}{dt}$  at any time t. q is given by the differential equation.

$$L \frac{d^2 q}{dt^2} + R \frac{dq}{dt} + \frac{q}{C} = E(t)$$

- The function sin(x) can be written as a Taylor series by:

$$\sin x = \sum_{k=0}^{\infty} \frac{(-1)^k x^{2k+1}}{(2k+1)!}$$

Write a user-defined function file that calculates sin(x) by using the Taylor series.

#### UNIT IV FLOW CONTROL AND FILES

6 hours

For – Loops, While – Loops, Break Statements, Logical Indexing, Pre allocation. Data Types: Introduction, Strings, Structs, Cells. Selection, If – Statements, Relational and Logical Operators, Nested If – Statements, Variables Number of Function, Arguments, Robustness, Persistent Variables. switch and case statement, while statement, break, Continue. **Files- File Input/ Output: File I/O, Excel Files, Text Files, Binary Files.**

- Without using the **max** command, find the maximum value of matrix (a) where a=[11 3 14;8 6 2;10 13 1]
- Let x=[2 6; 1 8], y=[.8 -0.3 ; -0.1 0.2], prove that y is not the inverse matrix of x
- The value of s could be calculated from the equation below:

$$s = \begin{cases} \sqrt{y^2 - 4xz} & \text{if } y \geq 4xz \\ \alpha & \text{if } y > 4xz \end{cases}$$

write a MATLAB program in M-File to do the following steps: -

- input the value of x, y, z
- calculate s
- print the output as shown below

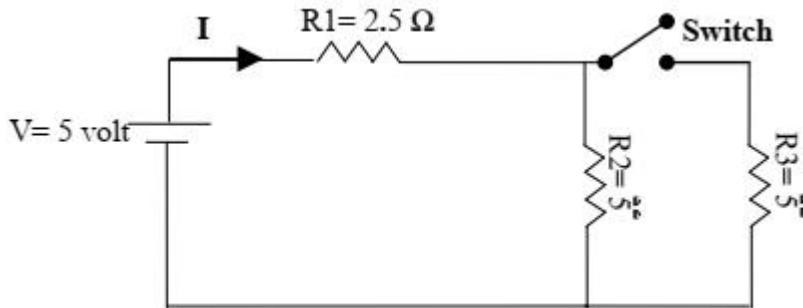
x = ...  
y = ...  
z = ...  
s = ...

- Use a for-end loop in a script file to calculate the sum of the first n terms of the series:

$$\sum_{k=1}^n \frac{(-1)^k k}{2^k}$$

Execute the script file for n = 4 and n = 20.

- Write a program to find the current I in the circuit shown below
  - a) By using conditional statements.
  - b) Without using any conditional statements.



## UNIT V PLOTTING AND SIMULINK

6 hours

**Two – Dimensional Plots** - Plot, fplot, Multiple Graphs, Formatting, Logarithmic Axes, Error Bars, Special Graphics, Histograms, Polar Plots, Multiple Plots on The Same Page, Multiple Figure Windows, **Three-Dimensional Plots**- Line Plots, Mesh and Surface Plots, Special Graphics, View Command. **Simulink**: Getting Started, Simulink Library Browser, Basic Elements-Blocks, Lines, building a System-Gathering Blocks, Modifying the Blocks, Connecting the Blocks, Running Simulations, Specification, Toolboxes, Building Systems.

- The expression for sine wave is given by  $x(t) = A \sin(2\pi ft + \phi)$ . Write a code which accepts the input as (i) Amplitude (A) (ii) Frequency (f) and (iii) Phase( $\phi$ ) and generates the sine wave. Plot the sine wave.
- Write a program to convert the sine wave to (i) Half wave rectified sine wave and (ii) Full wave rectified sine wave.
- Write a program which converts the sine wave to a square wave [Equivalent to that of “zero-crossing detector” or “comparator” concept in “Linear Integrated Circuits”].
- Write a program to generate three-phase sinusoidal signal. [The student should know what is the phase difference between three phases in a three-phase sinusoidal signal and the importance of three phase power]
- Design a Simulink block for power electronic circuits

### Course Outcomes:

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

1. Interpret the MATLAB commands and elementary functions
2. Solve mathematical operations and matrices manipulation

3. Apply MATLAB functions and expressions
4. Execute implementation of flow controls and files in MATLAB
5. Demonstrate Plotting and Simulink blocks in MATLAB.

**Text Books:**

1. Getting Started with MATLAB, Rudra Pratap Oxford University Press, 1<sup>st</sup> edition, 2019
2. MATLAB for Beginners: A Gentle Approach, Kattan, Peter Issa, Petra books, 2008

**Reference Books:**

1. MATLAB for Engineering Applications, William Palm, Mcgraw Hill, 4<sup>th</sup> edition, 2019.
2. MATLAB for Engineers, Holly Moore, Pearson Education, 5<sup>th</sup> edition, 2018

**Mode of Evaluation:** Continuous Internal Evaluation and End Semester Examination

# **DISCIPLINE ELECTIVE-I**

## **DISCIPLINE ELECTIVE-I**

### **MCA I Year II Semester**

#### **22MCAP401 MACHINE LEARNING – ALGORITHMS AND APPLICATIONS**

**L T P C**  
**3 0 0 3**

**Pre-requisite**          Algorithm Design & Programming, Probability & Statistics

#### **Course Description:**

This course provides a concise introduction to the fundamental concepts in machine learning and popular machine learning algorithms. This will cover the standard and most popular supervised learning algorithms including linear regression, logistic regression, decision trees, k-nearest neighbour, an introduction to Bayesian learning and the naïve Bayes algorithm, support vector machines and neural networks.

#### **Course Objectives:**

Students will be able to:

1. Know about overview of machine learning.
2. Learn about Regression and Classification methods.
3. Gain knowledge on Graphical models and Hidden Markov models.
4. Understand Clustering methods for real world problems
5. Learn neural networks for various inputs and expected outputs

#### **UNIT I          INTRODUCTION**

**9 hours**

Well posed learning problems, Designing a Learning system, Perspectives and issues in Machine Learning, Types of Machine Learning, Bayes theorem and Concept learning, Maximum likelihood and least squared Error hypotheses

#### **UNIT II          REGRESSION & CLASSIFICATION ALGORITHMS**

**9 hours**

Simple Linear Regression, Multiple Linear Regression, Classification Methods- Logistic regression, Nearest neighbor Classifier- Decision trees- Support Vector Machine, Genetic Algorithm.

#### **UNIT III          REINFORCEMENT ALGORITHMS**

**9 hours**

Reinforcement Learning – Components of Reinforcement, Types of Reinforcement, Applications, Reinforcement Learning Algorithms-Value-based, Policy-based, Model-based, Characteristics, Markov Decision Process, Q Learning.



**UNIT IV CLUSTERING ALGORITHMS****9 hours**

Clustering Methods-Partitioned based Clustering – K means, K medoids; Hierarchical Clustering – Agglomerative, Divisive, Distance measures; Density based Clustering, Gaussian model

**UNIT V ARTIFICIAL NEURAL NETWORK ALGORITHMS****9 hours**

Neural network representation, Perceptron, Multi Layer Perceptron with Back propagation, An Illustrative example: Face Recognition, Advance topics in Artificial Neural Networks

**Course Outcomes:**

After completion of the course the students will be able to:

1. Select real-world applications that needs machine learning based solutions.
2. Implement supervised Machine Learning algorithm for Classification and Regression.
3. Apply Reinforcement Modes for Real Time Predictions
4. Implement the appropriate Clustering real time algorithms for real world problems
5. Implement Artificial Neural Network for real time Applications.

**Text Book(s)**

Aurelien Geron, *Hands on Machine Learning with Scikit -Learn , Keras and Tensor Flow*

1. *Concepts, Tools and Techniques to build intelligent Systems*, 2<sup>nd</sup> Edition by O'Reilly Media Publishers, 2019.
2. Tom Michel , “Machine Learning”, Mc.Graw Hill, Indian Edition, 1997.

**Reference Books**

1. Mohri, Rostamizadeh and Talwalkar, “Foundations of Machine Learning”, MIT Press, 2012
2. Laura Igual Santi Seguí, *Introduction to Data Science, A Python approach to concepts and Applications*, Springer-2017.

**Mode of Evaluation:** Assignments, Mid Term Tests, End Semester Examination.

## MCA I Year II Semester

### 22MCAP402 IoT TECHNOLOGY AND APPLICATIONS

L T P C  
3 0 0 3

**Pre-requisite** Computer Network, Artificial Intelligence

#### **Course Description:**

Internet of Things is transforming our physical world into a complex and dynamic system of connected devices on an unprecedented scale. This course introduces the IoT technologies, protocols, and designing IoT systems with hardware basis. It provides basic information about Data Analytics and its Supporting Services. It imparts the Applications on industry, healthcare and smart cities.

#### **Course Objectives:**

Student will be able to:

1. Understand Smart Objects and IoT Architectures.
2. Identify the various IOT-related protocols
3. Create a simple IoT Systems using Arduino and Raspberry Pi.
4. Analyze the data analytics and cloud systems in the context of IoT
5. Classify the IoT infrastructure for popular applications.

#### **UNIT I FUNDAMENTALS OF IoT**

**9 hours**

Evolution of Internet of Things – Enabling Technologies – IoT Architectures: oneM2M, IoT World Forum (IoTWF) and Alternative IoT models – Simplified IoT Architecture- Core IoT Functional Stack– Smart Objects – Sensors, Actuators, Sensor Networks and Connecting Smart Objects

#### **UNIT II IoT PROTOCOLS**

**9 hours**

**IoT Access Technologies:** Physical and MAC layers, topology and Security of IEEE 802.15.4, 802.15.4g, 802.15.4e, 1901.2a, 802.11ah and LoRaWAN – Network Layer: IP versions, The Key Advantages and Adoption of the Internet Protocol, Constrained Nodes and Networks -Optimizing IP for IoT: 6LoWPAN, Header Compression, Fragmentation, Mesh Addressing, Routing over Low Power and Lossy Networks – **Application Transport Methods:** SCADA – IoT Application Layer Protocols: CoAP and MQTT.

#### **UNIT III BUILDING IoT WITH RASPBERRY PI AND ARDUINO**

**9 hours**

Design Methodology – Embedded computing logic - Building IOT with RASPBERRY PI- IoT Systems– Logical Design using Python – IoT Physical Devices & Endpoints - IoT Device -Building blocks -Raspberry Pi -Board - Linux on Raspberry Pi - Raspberry Pi Interfaces -Programming Raspberry Pi with Python- IoT Design and Prototyping Using Arduino Boards.

**UNIT IV DATA ANALYTICS AND SUPPORTING SERVICES 9 hours**

**Data Analytics for IoT:** Structured Vs Unstructured Data, Data in Motion Vs Data in Rest – Role of Machine Learning – No SQL Databases – Hadoop Ecosystem – Apache Kafka, Apache Spark, lambda – Edge Streaming Analytics and Network.

**UNIT V APPLICATIONS AND CASE STUDIES 9 hours**

**Applications:** Smart and Connected Cities-Underlying Technologies for Smart Cities, Driverless Vehicles, Smart Buildings Smart Campuses, Smart Grid- Health Care- Agriculture: Crop Water Stress Index, IoT Irrigation System, Crop Disease and Pest Management

**Case Study:** Smart City Streetlights Control and Monitoring

**Course Outcomes:**

After completion of the course, students will be able to

1. Explain the term 'internet of things' in different contexts.
2. Analyze various protocols for IoT.
3. Design an IoT system using Raspberry Pi/Arduino
4. Apply data analytics and use cloud offerings related to IoT.
5. Analyze applications of IoT in real time scenario.

**Text Book(s)**

1. IoT Fundamentals: Networking Technologies, Protocols and Use Cases for Internet of Things, David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Rob Barton and Jerome Henry, Cisco Press, 2017.
2. Hassan, Qusay F., ed. Internet of things A to Z: Technologies and Applications. John Wiley & Sons, 2018.

**Reference Books**

1. Arshdeep Bahga, Vijay Madiseti, “Internet of Things – A hands-on approach”, Universities Press, 2015.
2. Dieter Uckelmann, Mark Harrison, Michahelles, Florian (Eds), “Architecting the Internet of Things”, Springer, 2011.
3. Olivier Hersent, David Boswarthick, Omar Elloumi , —The Internet of Things – Key applications and Protocols, Wiley, 2012
4. Jan Ho` ller, Vlasios Tsiatsis , Catherine Mulligan, Stamatis , Karnouskos, Stefan Avesand. David Boyle, “From Machine-to-Machine to the Internet of Things – Introduction to a New Age of Intelligence”, Elsevier, 2014.
5. Dieter Uckelmann, Mark Harrison, Michahelles, Florian (Eds), —Architecting the Internet of Things, Springer, 2011.
6. Michael Margolis, Arduino Cookbook, Recipes to Begin, Expand, and Enhance Your Projects, 2nd Edition, O’Reilly Media, 2011.
7. Kamal, Raj. Internet of Things- Architecture and Design Principles. McGraw-Hill Education, 2017.

**Mode of Evaluation:** Assignments, Mid Term Tests, End Semester Examination.

**MCA I Year II Semester**

**22MCAP403 AGILE SOFTWARE DEVELOPMENT PROCESS**

**L T P C**  
**3 0 0 3**

**Pre-requisite** Software Engineering

**Course Description:**

Object oriented analysis, UML, Agile software development and Frameworks of Agile development. DevOps is fundamental for developers and application oriented personalities. Improved workflows and faster deployment will be helpful for developers in their project implementation.

**Course Objectives:**

Students will be able to:

1. Explain the object oriented techniques used in the real world software industries.
2. Acquire knowledge about the classes, objects and UML diagrams.
3. Compare agile with traditional software development and Contrast different agile methodologies.
4. Analyze the DevOps relationship to Agile, Lean and ITSM.
5. Improve workflows and speed up deployment.

**UNIT I INTRODUCTION TO OBJECT ORIENTED ANALYSIS 9 hours**

**Introduction:** SDLC & its variations, Approaches to system development, SSAD, OOAD, Unified Process and its characteristics, Unified Process life cycle, Basics of Object Orientation- OO development and its themes; Modelling and usefulness of OO development

**UNIT II OBJECT ORIENTED MODELING AND UML 9 hours**

**Object Oriented Modeling-**Capturing, modeling, documenting system requirements, and Object oriented basic components; Dynamic Modeling- Use-case Modeling and Various UML diagrams.

**UML:** Introduction to UML, UML Building Blocks. Conceptual model of UML, Architecture, Classes, Relationships, Common Mechanisms, Class-Object-Sequence-Activity-Use Case Diagrams.

**UNIT III INTRODUCTION TO AGILE AND ITS SIGNIFICANCE 9 hours**

Software Development Process. Iterative and Evolutionary Methods. Agile software development Vs other traditional methods. Agile Manifesto: Principles, and Benefits of agile development, User Stories, Generating User Stories. Agile Story: Evolutionary delivery, Scrum Demo, Planning game, Sprint back log, adaptive planning. Agile Motivation Problems with the Waterfall - Research Evidence; Agile roles

**UNIT IV AGILE FRAMEWORK****9 hours**

Method Overview, Life cycle phases and Work product roles and practices- Scrum, Extreme Programming., Unified process, EVO

**UNIT V INTRODUCTION OF DEVOPS****9 hours**

Continuous Delivery, and the three ways The Principles of Flow Feedback Continual Learning and Experimentation. Selecting which value stream to start with Understanding the Work in our value stream Making it visible Expanding - Design Organization and architecture Outcomes by Integrating Operations.

**Course Outcomes:**

After completion of the course, students will be able to

1. Analyze the object oriented techniques.
2. UML diagrams for real-world project components.
3. Identify the significance of agile development over the traditional approaches.
4. To examine the different frameworks of agile software development approaches.
5. To analyze DevOps and its relationship with agile development.

**Text Book(s)**

1. Object Oriented Analysis and Design with Applications by Grady Booch Robert, Third Edition.
2. Agile Software Development Best Practices for Large Software Development Projects by Thomas Stober, and Uwe Hansmann.
3. Shore, James. The Art of Agile Development: Pragmatic guide to agile software development. "O' Reilly Media, Inc.", 2007.
4. The DevOps Handbook by Gene Kim, Jez Humble, Patrick Debois and John Willis.

**Reference Books**

1. Object - Oriented Modeling and Design with UML 2nd Edition.
2. Agile Software Development, Principles, Patterns, and Practices by Robert C. Martin
3. Effective DevOps, by Jennifer Davis, Ryn Daniels.
4. DevOps for Web Development by MiteshSoni.

**Mode of Evaluation:** Assignments, Mid Term Tests, End Semester Examination.

## MCA I Year II Semester

### 22MCAP404 COMPUTER GRAPHICS AND MULTIMEDIA

L	T	P	C
3	0	0	3

**Pre-requisite** NIL

#### Course Description:

Computer Graphics and Multimedia is to teach students theory, technology, procedures, and skills in computer graphics and multimedia. The compulsory courses are intended for improvement of both basic and in-depth knowledge of image synthesis in computer graphics, speech processing and recognition, sound and video sequences algorithms for multimedia, and development of systems for human-computer interaction.

#### Course Objectives:

1. To provide knowledge and understanding in the fundamental principles of Computer Graphics and Mathematical concepts related to Computer graphical operations.
2. To provide in-depth knowledge of display systems, image synthesis and shape modelling of 3D applications.
3. To understand the basic concepts related to Multimedia including data standards, algorithms and software.
4. To Experience the development of Multimedia application to display their ability by using Multimedia tools.

#### UNIT I BASIC CONCEPTS

9 hours

2D Transformations – Clipping – Point Clipping – Line Clipping – Polygon Clipping – Text Clipping – Exterior Clipping – Window to View Port Mapping – Interactive Input Methods – Picture Construction Techniques.

#### UNIT II 3D GRAPHICS

9 hours

3D Concepts – 3D Transformations – 3D Viewing – Visible Surface Detection Methods – Back Face Detection Method – Depth Buffer Method – Scan Line Method – Virtual Reality Environment.

#### UNIT III MULTIMEDIA BASICS

9 hours

Introduction to Multimedia – Applications– Hypermedia – Authoring — File formats –Color Models – Digital Audio– Digital Music Making – MIDI – Digital Video – Video Compression Techniques – Video Performance Measurements –Multimedia Databases–Animation.

#### UNIT IV MULTIMEDIA COMMUNICATION

9 hours

Multimedia Network Services–Network Protocols–Requirements for Multimedia Communications – Multimedia Conferencing Architectures –QuickTime Movie File Format– MHEG–Multimedia File Sharing –Multimedia & Internet–Real-Time Interchange.

## **UNIT V      MULTIMEDIA APPLICATION DEVELOPMENT**

9 hours

Design of a Multimedia System –Content Based Information Retrieval – HDTV, ATV, EDTV, IDTV Standards –Development of User Interface Design – Multimedia Broadcasting –Social Media Sharing – Multimedia Development Issues – Sample Multimedia Project.

### **Course Outcomes:**

After successful completion of the course, students will be able to

1. Gain proficiency in various algorithms of 2D Computer graphics and trend their use in various real-life systems.
2. Enhance the perspective of Modern computer system with modelling, analysis and interpretation of 3D visual information.
3. Able to understand different forms of Multimedia and gain knowledge about Audio and Video.
4. Able to understand the Networks used for Multimedia and to communicate with Multimedia Applications.
5. Able to design and implement a number of Multimedia Applications and to do Research in Multimedia Industry.

### **Text Book(s)**

1. Donald Hearn and M. Pauline Baker, “Computer Graphics C Version”, Second Edition, Pearson Education.
2. David Hillman, “Multimedia – Technology and applications”, Galgotia Publications, Delhi, 2008.

### **Reference Books**

1. Ralf Steinmetz and Klara “Multimedia Computing, Communications and Applications”, Pearson Education, 2009
2. Tom McReynolds – David Blythe, “Advanced Graphics Programming Using OpenGL”, Elsevier, 2005

**Mode of Evaluation:** Assignments, Mid Term Tests, End Semester Examination.

MCA I Year II Semester

**22MCAP405 IMAGE PROCESSING**

**L T P C**

**3 0 0 3**

**Pre-requisite** NIL

**Course Description:**

This course is designed to provide basic understanding on image processing. The course material further used for developing any digital image processing applications. Course covers image enhancement methods, image restoration techniques, multi-resolution analysis, image segmentation models, feature extraction methods, supervised, unsupervised algorithms and applications.

**Course Objectives:**

Student will be able to:

1. Understand the basic concepts of digital image processing.
2. Understand the image enhancement techniques.
3. Expose the various image processing techniques and their applications.
4. Expose the use of current technologies in image processing systems.
5. Apply real world applications of image processing.

**UNIT I FUNDAMENTALS OF IMAGE PROCESSING**

**9 hours**

Introduction – Steps in Digital Image Processing – Components – Elements of Visual Perception – Image Sensing and Acquisition – Image Sampling and Quantization – Relationships between pixels and color models – Image operations.

**UNIT II IMAGE ENHANCEMENT**

**9 hours**

Image Transforms: Discrete Fourier Transform – Fast Fourier Transform – Discrete Cosine Transform – Image Enhancement in Spatial and Frequency Domain – Grey Level Transformations – Histogram Processing – Spatial Filtering – Smoothing and Sharpening – Frequency Domain: Filtering in Frequency Domain.

**UNIT III IMAGE RESTORATION AND MULTI-RESOLUTION ANALYSIS**

**9 hours**

Multi Resolution Analysis: Image Pyramids – Multi Resolution Expansion – Wavelet Transforms – Image Restoration – Image Degradation Model – Noise Modelling – Blur – Order Statistic Filters – Image Restoration Algorithms.



**UNIT IV IMAGE SEGMENTATION AND FEATURE EXTRACTION 9 hours**

Image Segmentation – Detection of Discontinuities –Edge Operators –Edge Linking and Boundary Detection – Thresholding – Region based Segmentation – Image Features and Extraction – Image Features – Types of Features – Feature Extraction – SIFT, SURF and Texture – Feature Reduction Algorithms.

**UNIT V IMAGE PROCESSING APPLICATIONS 9 hours**

Image Classifiers – Supervised Learning – Support Vector Machines, Image Clustering – Unsupervised Learning – Hierarchical and Partition based Clustering Algorithms – EM Algorithm

**Course Outcomes:**

After successful completion of the course, students will be able to

1. Understand basic image processing operations.
2. Apply new techniques in the areas of image enhancement and restoration.
3. Understand the image segmentation algorithms.
4. Extract features from images by using classifiers and clustering algorithms.
5. Design image processing application that uses different concepts of image processing.

**Text Book(s)**

1. Rafael Gonzalez, Richard E. Woods, “Digital Image Processing”, Fourth Edition, Pearson Education, 2018.
2. Milan Sonka, Vaclav Hlavac, Roger Boyle, “Image Processing Analysis and Machine Vision”, Fourth Edition, Cengage India, 2017.

**Reference Books**

1. S. Annadurai, S. Shanmugalakshmi, “Fundamentals of Digital Image Processing”, Pearson, 2006.
2. Anil K. Jain, “Fundamentals of Digital Image Processing”, PHI, 2011.

**Mode of Evaluation:** Assignments, Mid Term Tests, End Semester Examination.

**MCA I Year II Semester**

**22MCAP406 BLOCKCHAIN TECHNOLOGY**

**L T P C**  
**3 0 0 3**

**Pre-requisite:** 22MCAP105

**Course Description:**

This course is designed for the understanding of blockchain technology and cryptocurrencies. Upon completion of the programme students will be able to develop secure applications using blockchain concepts, design and develop automated smart contracts, implementation of industry-specific blockchain models and frameworks. Possible career opportunities upon completion of the programme are Blockchain business analyst, Blockchain Consultant, Blockchain Solution Architect, Cryptocurrency Developer, etc.

**Course Objectives:**

The course is designed to meet the objectives of:

4. To introduce the concept of Blockchain
5. To overcome the problems of centralization
6. To introduce the concept of Bitcoin
7. To make them familiar with Bitcoin network, payments, clients and APIs

**UNIT I INTRODUCTION TO BLOCKCHAIN 9 hours**

Introduction to Blockchain, How Blockchain works, Blockchain vs Bitcoin, Practical applications, public and private key basics, pros and cons of Blockchain, Myths about Bitcoin.

**UNIT II BLOCKCHAIN 9 hours**

Architecture, versions, variants, use cases, Life use cases of blockchain, Blockchain vs shared Database, Introduction to cryptocurrencies, Types, Applications.

**UNIT III CONCEPT OF DOUBLE SPENDING 9 hours**

Concept of Double Spending, Hashing, Mining, Proof of work. Introduction to Merkel tree, Privacy, payment verification, Resolving Conflicts, Creation of Blocks.

**UNIT IV INTRODUCTION TO BITCOIN 9 hours**

Introduction to Bitcoin, key concepts of Bitcoin, Merits and De Merits Fork and Segwits, Sending and Receiving bitcoins, choosing bitcoin wallet, Converting Bitcoins to Fiat Currency.

## **UNIT V INTRODUCTION TO ETHEREUM**

**9 hours**

Introduction to Ethereum, Advantages and Disadvantages, Ethereum vs Bitcoin, Introduction to Smart contracts, usage, application, working principle , Law and Regulations. Case Study.

### **Course Outcomes:**

After successful completion of the course, students will be able to

1. Demonstrate the basics of Block chain concepts using modern tools/technologies (L3).
2. Analyze the application of specific block chain architecture for a given problem (L4).
3. Evaluate the usage of Block chain implementation/features for the given problem (L5).
4. Exemplify the usage of bitcoins and its impact on the economy (L3).
5. Understand the basic concepts of Ethereum and its regulations (L2).

### **Text Book(s)**

1. Beginning Blockchain: A Beginner's Guide to Building Blockchain Solutions by ArshdeepBikramaditya Signal, GautamDhameja (PriyansuSekhar Panda., APress.
2. Blockchain Applications: A Hands-On Approach by Bahga, Vijay Madisetti
3. Blockchain by Melanie Swan, OReilly

### **Reference Books**

1. Bitcoin and Cryptocurrency Technologies by Aravind Narayan. Joseph Bonneau, princeton
2. Bitcoin and Blockchain Basics: A non-technical introduction for beginners by Arthu.T Books.

**Mode of Evaluation:** Assignments, Mid Term Tests, End Semester Examination.

**MCA I Year I Semester**

**22MCAP407 XML AND WEB SERVICES**

**L T P C**  
**3 0 0 3**

**Pre-requisite** Web designing concepts, HTML concepts

**Course Description:**

This course provides XML technology family, architecture of web services, the building blocks of Web services, the role of XML in E-business and content management.

**Course Objectives:**

Student will be able to:

1. Understand the overview of XML family and related technologies
2. Learn the concepts of Web Services Architecture
3. Understand the ideas on Web services building blocks
4. Identify the implementation of XML in E-Business
5. Understand the concepts of content management in XML

**UNIT I XML TECHNOLOGY FAMILY 9 hours**

XML – benefits – Advantages of XML over HTML – EDL –Databases – XML based standards – DTD –XML Schemas – X- Files – XML processing – DOM –SAX presentation technologies – XSL – XFORMS – XHTML – voice XML – Transformation – XSLT – XLINK – XPATH –XQ

**UNIT II ARCHITECTING WEB SERVICES 9 hours**

Business motivations for web services – B2B – B2C- Technical motivations – limitations of CORBA and DCOM – Service – oriented Architecture (SOA) – Architecting web services – Implementation view – web services technology stack – logical view – composition of web services – deployment view – from application server to peer to peer – process view – life in the runtime

**UNIT III WEB SERVICES BUILDING BLOCK 9 hours**

Transport protocols for web services – messaging with web services – protocols – SOAP – describing web services – WSDL – Anatomy of WSDL – manipulating WSDL – web service policy – Discovering web services – UDDI – Anatomy of UDDI- Web service inspection – Ad-Hoc Discovery – Securing web services.

**UNIT IV IMPLEMENTING XML IN E-BUSINESS 9 hours**

B2B - B2C Applications – Different types of B2B interaction – Components of ebusiness XML systems – ebXML – Rosetta Net Applied XML in vertical industry – Web services for mobile devices

**UNIT V XML AND CONTENT MANAGEMENT 9 hours**

Semantic Web – Role of Meta data in web content – Resource Description Framework – RDF schema – Architecture of semantic web – content management workflow – XLANG –WSFL.

**Course Outcomes:**

After completion of the course, students will be able to

1. Illustrate the ideas on XML concepts
2. Explain the architecture of web services.
3. Analyze web designing building blocks
4. Explain the implementation ideas XML in E-Business
5. Illustrate the XML content management system

**Text Book(s)**

1. Ron schmelzer et al, “XML and Web Services”, Pearson Education, 2002.
2. Sandeep Chatterjee and James Webber, “Developing Enterprise Web Services: An Architect’s Guide”, Prentice Hall, 2004

**Reference Books**

1. Henry Bequet and Meeraj Kunnumpurath, “Beginning Java Web Services”, Apress, 2004
2. Russ Basiura and Mike Batongbacal, “Professional ASP.NET Web Services”, Apress, 2003.

**Mode of Evaluation:** Assignments, Mid Term Tests, End Semester Examination.