Tripura University

Department of Information Technology

Curriculum Structure

2-year Master of Computer Applications (MCA)

First Semester Examination, December
Second Semester Examination, April/May
Third Semester Examination, December
Fourth Semester Examination, April/May

Tripura University
A Central University, Suryamaninagar
Tripura West (799022)

Master of Computer Applications

Features:

1. Study through Core subjects, flexible and diverse program specific electives.
2. Open Electives to widen computer application based knowledge.
3. Foundation compulsory course.
4. Engagement of Industry in developing computer applications and solutions.
5. Ensured competency development of learner.

*Students to be encouraged to go to Industrial Training#/Internship for at least 2-3 months duringsemester break.

 # Students going for Industrial Project/ Thesis will complete these courses through MOOCs in the final semester.

Program Outcomes (PO) of MCA Program:

1. Apply knowledge of Computing fundamentals, specialization, Mathematics, and domain knowledge appropriate for the computer application to the abstraction and conceptualization requirements.
2. Identify, formulate, research literature, and solve computer and research in relevant domains.
3. Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of information to provide valid conclusions.
4. Create, select, adapt and apply appropriate techniques, resources, and modern computing tools to latest computing activities, with an understanding of the limitations.
5. Understand and commit to professional ethics and cyber regulations, responsibilities, and norms of professional computing practice.
6. Recognize the need, and have the ability to engage in independent learning for continual development.
7. Demonstrate knowledge and understanding of computing and management principles and apply as a member and leader in a team, to manage projects in multidisciplinary environments.
8. Communicate effectively with society at large, being able to comprehend and write effective reports, design documentation, make effective presentations, and give and understand clear instructions.
9. Understand and assess social, environmental, health, safety, legal, and cultural issues within local and global contexts, and the consequential responsibilities relevant to professional practice.
10. Identify a timely opportunity and using innovation to pursue that opportunity to create value and wealth for the betterment of the individual and society at large.

Program Specific Outcomes (PSO) of MCA program:

The programme will enable the student to:

1. To prepare graduates who will create systems through software development to solve problems in Industry domain areas.
2. To Prepare Graduates who will contribute to societal growth through research in their chosen field.
3. To prepare graduates who will perform both as an individual and in a team through good analytical, design and implementation skills.
4. To prepare graduates who will be lifelong learners through continuous professional development.

Examination Schedule:

1. First Semester Examination, December- Every year
2. Second Semester Examination, April/May- Every year
3. Third Semester Examination, December- Every year
4. Fourth Semester Examination, April/May- Every year

2-year Master of Computer Applications
Detail Syllabus (CBCS Pattern) AY 2020-21
Curriculum Structure

MCA SEMESTER I

Course Code	Course Title	L-T-P	Credits	Hours	Mark	MOOC
MCA0101C	Mathematical Foundations of Computer Applications	3-1-0	4	4	100	Yes*
MCA0102C	Programming in C	3-1-0	4	4	100	Yes*
MCA0103C	Computer Organization & Assembly Language Programming	3-1-0	4	4	100	Yes*
CSK III	Computer Skill-III	4-0-0	4	4	100	Yes
MCA0104C	Programming Laboratory	0-0-4	2	4	100	N/A
MCA0105C	Assembly Language Laboratory	0-0-4	2	4	100	N/A
Total Credits	***4 Theory, 2 Laboratories***	*13-3-8*	***20***	***24***	***600***	

MCA SEMESTER II

Course Code	Course Title	L-T-P	Credits	Hours	Mark	MOOC
MCA0201C	Software Engineering	3-1-0	4	4	100	Yes*
MCA0202C	Data Structures & Algorithm	3-1-0	4	4	100	Yes*
MCA0203C	Operating System	3-1-0	4	4	100	Yes*
MCA0204C	Data Structures& Algorithm Laboratory	0-0-4	2	4	100	N/A
MCA0205C	Software Development Laboratory	0-0-4	2	4	100	N/A
MCA0206C	Seminar & Technical Writing	0-0-8	4	8	100	N/A
Total Credits	***4 Theory, 3 Laboratories***	*9-3-16*	***20***	***28***	***600***	

MCA SEMESTER III

Course Code	Course Title	L-T-P	Credits	Hours	Mark	MOOC
MCA0301C	Database Management Systems	3-1-0	4	4	100	Yes*
MCA0302C	Data Communication & Computer Network	3-1-0	4	4	100	Yes*
MCA00XXE	Elective I	4-0-0	4	4	100	Yes*
MCA00XXE	Elective II	4-0-0	4	4	100	Yes*
MCA0303C	Database Management Systems Laboratory	0-0-4	2	4	100	N/A
MCA0304C	Computer Network Laboratory	0-0-4	2	4	100	N/A
MCA0305C	Project Phase I	0-0-4	2	4	100	N/A
Total Credits	***4 Theory, 3 Laboratories***	***14-2-12***	***22***	***28***	***700***	

MCA SEMSTER IV

Course Code	Course Title	L-T-P	Credits	Hours	Mark	MOOC
MCA0401C	Project and Viva Voce	0-0-20	10	20	400	N/A
MCA00XXE	Elective III	4-0-0	4	4	100	Yes*
MCA00XXE	Elective IV	4-0-0	4	4	100	Yes*
Total Credits	***2 Theory,1 Laboratories***	**8-0-20**	**18**	**28**	**600**	

Total Credit= 80, Core Credit= 56 (Theory=32, Practical=24), Foundation=4, Elective= 20 (Departmental Elective= 16, Non-Departmental Elective= 04)

NB: If Semester-IV Project done outside department (in Industry) the Elective III and IV may be completed via taking extra electives in Semester III or Credit Transfer from MOOCs as per TU norms

Yes*: If available online in the particular semester but as per TU credit transfer norms

ELECTIVE SUBJECTS (DEPARTMENTAL) (16 Credits) to be completed in 2nd year (Semester III/IV)

Course Code	Course Title	L-T-P	Credits	MOOC
MCA0001E	Adhoc & Sensor Networks	4-0-0	4	Yes*
MCA0002E	Advanced Networking	4-0-0	4	Yes*
MCA0003E	Advances in Database	4-0-0	4	Yes*
MCA0004E	Artificial Intelligence	4-0-0	4	Yes*
MCA0005E	Cloud Computing	4-0-0	4	Yes*
MCA0006E	Cryptography and Network Security	4-0-0	4	Yes*
MCA0007E	Data Mining	4-0-0	4	Yes*
MCA0008E	Data Science	4-0-0	4	Yes*
MCA0009E	Deep Learning	4-0-0	4	Yes*
MCA0010E	Digital Logic & Basic Electronics	4-0-0	4	Yes*
MCA0011E	Digital Signal Processing	4-0-0	4	Yes*
MCA0012E	Discrete Mathematical Structures	4-0-0	4	Yes*
MCA0013E	Distributed Computing	4-0-0	4	Yes*
MCA0014E	Formal Language and Automata Theory	4-0-0	4	Yes*
MCA0015E	Image Processing	4-0-0	4	Yes*
MCA0016E	Information Retrieval and Web Mining	4-0-0	4	Yes*
MCA0017E	Internet of Things	4-0-0	4	Yes*
MCA0018E	Internet Technology	4-0-0	4	Yes*
MCA0019E	Machine Learning	4-0-0	4	Yes*
MCA0020E	Multimedia Technology	4-0-0	4	Yes*
MCA0021E	Natural Language Processing	4-0-0	4	Yes*
MCA0022E	Network Synthesis	4-0-0	4	Yes*
MCA0023E	Numerical Methods	4-0-0	4	Yes*
MCA0024E	Object Oriented Analysis and Design	4-0-0	4	Yes*
MCA0025E	Pattern Recognition	4-0-0	4	Yes*
MCA0026E	Social Networks	4-0-0	4	Yes*
MCA0027E	Soft Computing	4-0-0	4	Yes*
MCA0028E	Software Project Management	4-0-0	4	Yes*
MCA0029E	Speech and Natural Language Processing	4-0-0	4	Yes*
MCA0030E	TCP/IP Network Programming	4-0-0	4	Yes*
MCA0031E	Web Technology	4-0-0	4	Yes*
MCA0032E	Information System Security	4-0-0	4	Yes*
MCA0033E	Advanced SoC Design	4-0-0	4	Yes*
MCA0034E	Advanced Computer Architecture	4-0-0	4	Yes*
MCA0035E	Embedded Linux	4-0-0	4	Yes*
MCA0036E	Graphics and Mobile Gaming	4-0-0	4	Yes*
MCA0037E	Introduction to Robotics Systems	4-0-0	4	Yes*
MCA0038E	Embedded Systems Design	4-0-0	4	Yes*
MCA0039E	Object Oriented Programming	4-0-0	4	Yes*
MCA0040E	Programming in Python	4-0-0	4	Yes*

- **Elective Subjects (Departmental): 16 Credits, Open Elective (Non-Departmental) Subjects: 4 Credits.**
- **Yes*: If available online in the particular semester but as per TU credit transfer norms.**

First Semester

Mathematical Foundations of Computer Applications	**MCA0101C**
3 - 1 - 0 : 4 Credits	Prerequisites: *None*

Course Outcomes

At the end of the Course student will have

- Ability to apply mathematical logic to solve problems. Understand sets, relations, functions, and discrete structures.
- Ability to use logical notation to define and reason about fundamental mathematical concepts such as truth table, DeMorgan's Law and Predicate Calculus.
- Ability to formulate problems and solve Determinants and Matrices.
- Ability to model and solve real-world problems using Statistics and Probabilities.

Syllabus

Set Theory and Relations: Set, Properties, sub-set, Equality of sets, Union, Intersection and Complements, Symmetric Difference, De Morgan's Law, Cartesian Product, Binary Relations, Principles of Inclusions and Exclusions, Generating Function, Recurrence relation, Venn Diagram. The Pigeonhole Principle (Definition and Problem), Product Sets and Partition, Relations and their Properties and Representation, Equivalence of Relations, Manipulation of Relations, Graph Theory.

Discrete Mathematics: Proposition, Truth value, Truth tables, Connectives, Negation, Conjunction, Disjunction, Implication, Combination of Connectivity's, Converse, Inverse and Contra positive of an Implication, Tautology, Logical Equivalence, Idempotent, Associative-, Commutative-, Distributive and De Morgan's Laws, Disjunctive and Conjunctive Normal Forms, Predicate Calculus

Linear Algebra: Vector algebra and calculus, Determinants, Matrices, Solution of linear algebraic equations, Cayley-Hamilton Theorem, Linear differential equations of second and higher order, Partial differentiation and Partial difference equations, The expected difference operator and solution using E-operator

Statistics and Probability: Concept of mean, medial, and mode, moments about mean, variance, skewness, kurtosis, Classical definition of probability, probability density functions, conditional and marginal probabilities, expectation, Bayes' theorem, Binomial, Poisson, Normal and Gaussian distributions, Hypothesis testing

Textbooks:

1. "Mathematical Foundations Of Computer Science" – G. Shankar Rao
2. "Discrete Mathematical Structures With Applications To Computer Science" – Jean-Paul Tremblay and R. Manohar
3. "Engineering Mathematics"- Kryzig

References:

1. "Discrete mathematics Structures" – Bernard Kolman, Robert C. Busby,Sharoncutler Ross
2. "Discrete mathematics" - Olympia Nicodemi.
3. "Elements of Discrete Mathematics" - C. L. Liu
4. "Introduction to Numerical Analysis, Probability and Statistics" - A. K. Mukhopadhay

Programming in C	**MCA0102C**
3 - 1 - 0: 4 Credits	Prerequisites: *None*

Course Outcomes

At the end of the Course student will be able to

- Implement the algorithms and draw flowcharts for solving Mathematical and Engineering problems. Understanding of computer programming language concepts.
- Ability to design and develop Computer programs, analyzes, and interprets the concept of pointers, declarations, initialization, operations on pointers and their usage.
- Able to define data types and use them in simple data processing applications also be able to use the concept of array of structures.
- Define union and enumeration user defined data types. Develop confidence for self-education and ability for life-long learning needed for Computer language.

Syllabus

Introduction to Problem Solving through programs, Flowcharts/Pseudo codes, The compilation process, Syntax and Semantic errors, Variables and Data Types, Arithmetic expressions, Relational Operations, Logical expressions; Introduction to Conditional Branching;
Conditional Branching and Iterative Loops
Arrays: 2-D arrays, Character Arrays and Strings
Basic Algorithms including Numerical Algorithms
Functions and Parameter Passing by Value
Passing Arrays to Functions, Call by Reference; Recursion
Structures and Pointers; Self-Referential Structures and Introduction to Lists

Textbooks:

1. Byron Gottfried, Schaum's Outline of Programming with C, McGraw-Hill
2. E. Balaguruswamy, Programming in ANSI C, Tata McGraw-Hill

Reference Books:

1. Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, Prentice Hall of India

Computer Organization & Assembly Language Programming	**MCA0103C**
3 - 1 - 0: 4 Credits	Prerequisites: *None*

Course Outcomes

At the end of the Course student will be able to

- Discuss the architecture of 8085 processors, instruction sets and timing diagram.
- Have the concept of Computer Architecture along with micro and macro programming.
- Understand various interrupts and the concept of interfacing.
- Understand the basics of different processors and their architecture

Syllabus

Computer: Definition, Block Diagram, Organizations architecture, Computer Evolution, Performance, History; Von Neumann Architecture, Designing for Performance, Computer Components, Computer Function, Bus Interconnection; Number System: Conversions and Operations, Data Representation; 1's and 2's Complement Arithmetic, Floating Point Number Arithmetic. Boolean Algebra: Simplification of Boolean function using K-Map, Logic Gates, Binary Adder, Subtractor, Decoders, Encoders, Multiplexers, De-Multiplexers.

General Register Organization: Arithmetic and Logic Unit Organization; Organization of Central Processing Unit; Stack Organization; Memory Organization: Memory Hierarchy, Main Memory, Cache Memory, Virtual Memory, Memory Mapping Techniques; RISC & CISC Architectures and their Comparison.

Introduction to Microprocessor: Microcomputer and Assembly Language, Evolution of Microprocessor and their Application;Architecture and Organization of 8 bit microprocessor;
Intel 8085A Microprocessor- Pin Description, Functional block diagram; Registers, Flags; Interrupt structure; Instruction format, Instruction set: Classification, Execution; Addressing modes, Instruction Cycle, Timing diagram: op-code fetch machine cycle, memory read/write machine cycles, I/O read/write machine cycles;Simple Assembly Language Programming.

Programmable peripheral devices: Pin configuration and Architecture of 8237/8257 DMA Controller, 8253/8254 Programmable Interval Timer, 8255 Programmable Peripheral Interface, 8259 Programmable Interrupt Controllerand 8279 Programmable Keyboard/Display;

Text Books:

1. RomeshGaonkar, "Microprocessor Architecture, Programming and Application with the 8085", PRI publication
2. Senthil Kumar, " Microprocessors and Microcontroller", Oxford
3. Morris Mano, "Computer System Architecture", PHI

References:

1. Srinath, "8085 Microprocessor, programming and interface", PHI
2. B. Ram, "Computer Fundamental Architecture & Organization", New Age

Computer Skill-III	**CSK III**
4 - 0 - 0 : 4 Credits	Prerequisites: *Basic Programming and Logic*

Course Outcomes

At the end of the Course student will be able to

- Use an integrated development environment to write, compile, run, and test simple object-oriented Java programs.
- Read and make elementary modifications to Java programs that solve real-world problems.
- Validate input in a Java program. Identify and fix defects and common security issues in code.
- Use a version control system to track source code in a project.

Syllabus

Fundamentals of Object-Oriented Programming, Java Evolution, Java History
Java Features: Overview of Java Language, Constants, Variables and Data Types, Operators and Expressions, Decision making, branching and looping.
Classes, Objects and Methods, Arrays, String and Collections, Interfaces, Packages, Managing Errors and Exceptions
Multithreading, Applet Programming, Java AWT, Event Handling
Java I/O Handling, Java Database Connectivity.

Books/References:

1. Programming in Java by SachinMalhotra and SaurabhChoudhary, Oxford Higher education.
2. Java: The Complete Reference by Herbert Schildt, McGraw-Hill Education.
3. Java: A Premier by E. Balaguruswami, Tata Mcgraw Hill Education Private Limited.

Programming Laboratory	**MCA0104C**
0 - 0 - 4 : 2 Credits	Prerequisites: *Knowledge of Programming in C*

Course Outcomes

At the end of the Course student will be introduced to

- Familiarization of a computer and the environment (Editor, Compiler and Debugger); Execution of sample programs.
- Expression evaluation; Conditionals and Branching; Control Statements; Iteration; Functions; Recursion.
- Arrays; Pointer; Dynamic Memory Allocation; File Input Output; Numerical Methods; Searching & Sorting techniques.
- Advanced Topic based on the Programming Language used/ Object Oriented Programming topics.

Syllabus

Laboratory activities will be based on the syllabus of subject '**Programming in C**'. or any other Programming Language like Java (as per CSK III) or Python or C++.

Assembly Language Laboratory	**MCA0105C**
0 - 0 - 4 : 2 Credits	Prerequisites: *Knowledge of Assembly Language Programming*

Course Outcomes

At the end of the Course student will be introduced to

- Differentiating between High Level and Low Level Programming languages
- Understanding Assembly Level Programming and Mnemonics
- Implementing hardware programming using 8085 and other 8 bit microprocessors
- 16bit Programming and Higher order bit programming

Syllabus

Laboratory activities will be based on **Assembly Language Programming** part of subject 'Computer Organization & Assembly Language Programming'.

Second Semester

Software Engineering	**MCA0201C**
3 - 1 - 0 : 4 Credits	Prerequisites: *None*

Course Outcomes

At the end of the Course student will be introduced to

- Acquire strong fundamental knowledge in fundamentals of computer science, software engineering and multidisciplinary engineering to begin in practice.
- Design applicable solutions in one or more application domains using software engineering approaches that integrate ethical, social, legal and economic concerns.
- Deliver quality software products by possessing the leadership skills as an individual and demonstrating effective strategies by applying both communication and negotiation management skill.
- Application of software models, techniques and technologies to bring out innovative and novelistic solutions and professional development.

Syllabus

Introductory concepts: The evolving role of software – Its characteristics, components and applications- A layered technology – the software process – Software process models - Software process and project metrics – Measures, Metrics and Indicators- ethics for software engineers.

Software Project Planning: Project planning objectives – Project estimation – Decomposition techniques – Empirical estimation models - System Engineering- Risk management.
Analysis and Design – Design concept and Principles, Methods for traditional, Real time of object oriented systems – Comparisons – Metrics- Quality assurance

Testing fundamentals: Test case design – White box testing – Basis path testing – Control structure testing – Black box testing – Strategies: Unit testing integration testing – Validation Testing – System testing – Art of debugging – Metrics, Testing tools.

Formal Methods: Clean-room Software Engineering – Software reuse – Reengineering – Reverse Engineering – standards for industry

Text Books:

1. Rajib Mall, "Fundamentals of Software Engineering", 3rd Edition, PHI, 2009**.**
2. R. S. Pressman, "Software Engineering – A practitioner's approach", 5th Ed., McGraw Hill Int. Ed., 2001.

References:

1. Stephen R. Schach, "Classical & Object Oriented Software Engineering", IRWIN,1996.
2. James Peter, W. Pedrycz, "Software Engineering: An Engineering Approach", JohnWiley& Sons.
3. Ian Sommerville, Software engineering, 8thEdition,Pearson education Asia, 2007.
4. PankajJalote, "An Integrated Approach to Software Engineering", Springer Verlag,1997.

Data Structures& Algorithm	**MCA0202C**
3 - 1 - 0 : 4 Credits	Prerequisites: *None*

Course Outcomes

At the end of the Course student will be introduced to

- Ability to analyze algorithms and an algorithm correctness
- Ability to describe stack, queue and linked list operation
- Ability to have knowledge of tree and graphs concepts
- Ability to summarize searching and sorting techniques

Syllabus

Data Structures & Algorithm: Definition, Characteristics, Analysis of an Algorithms, Complexity, big O notation, Space Time trade-off; Data Structure - Definition, Classifications, Operations on data structure, Applications; Recursion - Definition, Properties, Algorithms for factorial and Fibonacci series.

Array & Linked List: Array - Definition, Representation, Multidimensional Arrays; Algorithms for insertion and deletion, Regular Array, Pointer Array, Jagged Array, Sparse matrix. Linked List - Definition, Representation, Types of Linked List, Algorithms for insertion and deletion, Applications.

Stack & Queue: Stack - Definition, Representation, Algorithms for push and pop, Conversion from infix to postfix and evaluation of postfix expression; Queue - Definition, Representation, Algorithms for insertion and deletion, Types of queue.

Searching & Sorting: Searching - Linear and Binary search with algorithms and complexity analysis; Hashing – hash functions, collision resolution techniques; Sorting - Definition, Bubble sort, Selection sort, Insertion sort, Quick sort, Merge sort; Radix sort; Heap sort with Complexity analysis;

Tree & Graphs: Tree - Definition; Representation; Binary Tree- Definitions and Properties, Binary tree traversal algorithms (pre, post, in), BST (Binary Search Tree) – creation, insertion and deletion, AVL tree, B-tree; B+ tree, B* tree, Application of trees; Graphs - Definitions, Representations; Breadth-first and Depth-first Search; Spanning Tree; Prim's and Kruskal's algorithms.

Text Books:

1. S.Lipschutz, "Theory and Problem of Data Structure", Schaum's Outline Series, Tata McGraw-Hill
2. Tannenbaum, "Fundamentals of Data Structures", PHI

References:

1. R.L. Kruse, B.P. Leary, C.L. Tondo, "Data structure and program design in C" , PHI
2. Horowitz and Sahani, "Fundamentals of Data structures", Galgotia publications
3. "Data Structures Using C" - ReemaThareja
4. "Introduction to Data Structures in C" – Ashok N. Kamthane
5. "Classic Data Structures" - D Samanta

Operating System	**MCA0203C**
3 – 1 – 0 : 4 Credits	Prerequisites: *None*

Course Outcomes

At the end of the Course student will be introduced to

- Analyze the structure and basic architectural components involved in OS
- Demonstrate competence in recognizing and using operating system features
- Understand and analyze theory and implementation of different operating system aspect
- Apply knowledge of different operating system algorithms

Syllabus

Introduction: Concept, Evolution, Functions; Single user, Multi-user and multitasking, Networked OS, Operating System structures: Monolithic, Layered, Virtual machine, OS services, System calls.

Processes & Threads: Concept, Process State and State Transitions, Process Control Block, Suspend & Resume of Process, Inter-process communication, Interrupt Processing, Process scheduling; Process Synchronization: critical-section Problem, Dekker's Algorithm, Semaphores, producer-consumer problem. Threads: Single & Multithreading Models, Threading issues, P threads, Solaris 2 Threads, Window 2000 Threads, Linux Threads, Java Threads.

Resource Management: Deadlocks: System Model, Characterization, Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance & Banker's Algorithm, Deadlock Detection, Deadlock Recovery. Memory Management: Memory Organization, Storage Hierarchy, Storage Management Strategies, Virtual memory: Paging, Segmentation, Segmentation with Paging, Page replacement algorithms. Device Management: I/O devices and subsystem, Polling, Interrupts; DMA: Principles and Operational details, Kernel I/O subsystem.

File-System & Security: File Concepts, File Organization, Access Methods, Directory Structure, File-system Mounting, File Sharing, Protection; File-system Structure, File System Implementation, Directory Implementation, Allocation Methods, Free-Space Management, Efficiency and Performance, Recovery. The security Problem, User Authentication, Program Threats, System Threats, Securing Systems and Facilities, Intrusion Detection, Cryptography, Computer-Security Classifications.

Text Books:

1. Operating System Concepts, Abraham Silberschatz, Peter Baer Galvin & Greg Gagne, John& Wiley & Sons, Inc.
2. Operating System, H M Deitel, Pearson Education, LPE.

References:

1. An Introduction to Operating System Concepts & Practice, Pramod Chandra P Bhatt, PHI Pvt Ltd.
2. An Introduction to Operating System, NIIT, PHI Pvt Ltd.

Data Structures and Algorithm Laboratory	**MCA0204C**
0 – 0 – 4 : 2 Credits	Prerequisites: *Data Structures*

Course Outcomes

At the end of the Course student will be introduced to

- Professional/academic knowledge and skills, Understand the properties of various data structures.
- Identify the strengths and weaknesses of different data structures.
- Design and employ appropriate data structures for solving computing problems.
- Possess the knowledge of various existing algorithms;Analyze and compare the efficiency of algorithms

Syllabus

Laboratory activities will be based on the syllabus of subject "Data Structures and Algorithm".

Software Development Laboratory	**MCA0205C**
0 – 0 – 4 : 2 Credits	Prerequisites: *Operating System, Software Engg*

Course Outcomes

At the end of the Course student will be

- Capable to acquire the generic software development skill through various stages of software life cycle.
- Able to ensure the quality of software through software development with various protocol based environment.
- Able to generate test cases for software testing, handle software development models through rational method.
- Able to design Modelling, UML Notation, Test cases, Test Suits, Rational Unified Process algorithms

Syllabus

Laboratory activities will be based on the syllabus of subject "Software Engineering" and Application development.

Seminar & Technical Writing	**MCA0206C**
0 - 0 - 8 : 4 Credits	Prerequisites: None

Course Outcomes

At the end of the Course student will be introduced to

- Research Skills
- Correspondence Skills
- Promotional Writing Skills
- Visual Communication Skills

Syllabus

Writing technical paper or research paper in ACM/ IEEE /Elsevier format.
Introduction to Technical Writing: how differs from other types of written communication Purpose of technical writing, Correspondence: prewriting, writing and rewriting Objectives of Technical Writing. Audience Recognition: High-tech audience, Low tech audience, Lay audience, Multiple Audience.
Correspondence: Memos, Letters, E-mails, Its differentiation, types of letters, Document Design, its importance, Electronic Communication: Internet, Intranet, extranet, Writing effective e-mail.
Summary: Report Strategies, Effective style of technical report writing: Structures: content, introduction, conclusions, references, etc., Presentation, Writing first draft, revising first draft, diagrams, graphs, tables, etc. report lay-out.
Report Writing: Criteria for report writing, Types of Report: Trip report, Progress report, lab report, Feasibility report, project report, incident report, etc. Case Studies.
Proposals & Presentation: Title page, Cover letter, Table of Content, list of illustrations, summary, discussion, conclusion, references, glossary, appendix, Case Studies. Oral Presentation/ Seminar:

References:

1. Sharon J. Gerson& Steven M. Gerson "Technical Writing – Process& Product", Pearson Education.
2. Sunita Mishra, "Communication Skills for Engineers" Pearson Education
3. Davies J.W. "Communication for engineering students", Longman Eisenberg, "Effective Technical Communication", Mc. Graw Hill.

Third Semester

Database Management Systems	**MCA0301C**
3 - 1 - 0 : 4 Credits	Prerequisites: *None*

Course Outcomes

At the end of the Course student will be introduced to

- An understanding of the needs for and uses of database management systems. Understanding of the context, phases and techniques for designing and building database systems.
- An understanding of the components of a computerized database information system (application).
- An ability to correctly use the techniques, components and tools of a typical database management system, such as Access or Oracle, to build a comprehensive database information system (application).
- An ability to design a correct, new database information system for a business functional area and implement the design, in either Access or Oracle.

Syllabus

Evolution of data management, File management vs. Data management, Components of a DBMS, DBMS Architecture, Data models.
Entity-Relationship Model, Conversion from E-R data model to Relational model
Relational Algebra, RelatinalCalculas and SQL; Normalizatin- Functional dependency, 1NF, 2NF, 3NF, BCNF, 4NF, 5NF
Basic File structure, Operation on file, Hashing techniques, Indexing structures, Primary and secondary indexing, Multilevel indexing using B tree.
Query Processing and Optimization-Optimization through algebraic manipulation. Cost based optimization. Join algorithm.
Transaction-ACID properties, Serializability-Conflict and View, Testing Serializability;
Concurrency control and Recovery technique- Two phase locking technique, Deadlock and stagnation based protocol; Log-based Recovery.
Object Oriented Data Bases: Approaches, Modeling and Design, Persistence, Transaction, Concurrency, Recovery;

Books Recommended

1. Database System Concepts, Abraham Silberschatz, Henry F. Korth, S. Sudarshan, The McGraw−Hill, Sixth edition.
2. Fundamentals of Database Systems, Dr. RamezElmasri, Shamkant B. Navathe, Fourth edition, Pearson.
3. Database Management Systems, Raghu Ramakrishnan, Johannes Gehrke, The McGraw−Hill, 3rd edition.

Data Communication & Computer Networks	**MCA0302C**
3 - 1 - 0 : 4 Credits	Prerequisites: *None*

Course Outcomes

At the end of the Course student will be introduced to

- Build an understanding of the fundamental concepts of computer networking.
- Familiarize the student with the basic taxonomy and terminology of the computer networking area.
- Introduce the student to advanced networking concepts, preparing the student for entry Advanced courses in computer networking.
- Allow the student to gain expertise in some specific areas of networking such as the design and maintenance of individual networks.

Syllabus

Data communication- Component, Data flow, Data representation, protocol and standards; Computer Network-Definition, Use, measuring a network - latency, data transfer rate, bandwidth; Types of network; Network model: Client Server and peer-to-peer model; Network Models;

Digital-to-Analog Conversion: ASK, FSK, PSK;Quadrature Amplitude Modulation; Analog-to-AnalogConversion: AM, FM, PM; Line Coding, Line Coding Schemes Block Coding;

Sampling theorem, Nyquist interval, PAM, PWM, PPM, Sampling and quantization, PCM, Encoder, Decoder, DPCM, Delta Modulation

Physical Layer-Analog and Digital signal of data, sine wave, Time and Frequency Domain; Digital Signal- Bit rate, Bit interval, Baud rate, Noise, Attenuation, Distortion, Throughput, Propagation speed, Propagation time, Wavelength.Transmission mode; Transmission media: Guided and unguided.

Data Link Layer-Errors: Single bit, Burst error; Error Detection- Redundancy, Parity check, CRC, Checksum; Flow and Error control mechanism. Stop-and-Wait ARQ, GO-BACK-N-ARQ, Sliding Window Protocol, Error detection mechanism, VRC,LRC,CRC, Automatic Repeat Request(ARQ)- stop-and wait , go-back-n, Selective Repeat-ARQ; HDLC- configuration and transfer modes, frames, frame format, frame type.

Medium Access Control- ALOHA, Slotted ALOHA, CSMA, CSMA/CD, Ethernet, Token Ring, CSMA/CA. Access method: point-to-point, multi point.

LAN- Traditional Ethernet, MAC Sub layer, Physical layer, Three generation of Ethernet. 10Base2, 10Base5, 10Base-T, 10Base-FL

Network Layer-Connection–oriented and connectionless services, Internetworking; Repeater, Hub, NIC, Switch, Bridge, Router, Gateway; Addressing: IP address, classful address, subnetting, supernetting, Classless Addressing; Routing techniques –Static, Dynamic, flooding, Distance vector and link-state routing, Basics of IP. Protocol- ARP, IPv4, ICMP

Transport Layer-Congestion control algorithm. Basics of TCP and UDP;

Application Layer- Basics of Protocols: DNS, TELNET, FTP, SMTP and MIME, HTTP HTML, CGI, WWW –Hypertext and Hypermedia; browser; static and dynamic documents.

Text Book:

1. Data Communication and Networking, 4th Edition, McGrawhill, Forouzan.

References:

1. A.Tanenbaum, “Computer Networks”, 4th Ed., Pearson Education Asia (LPE), 2003.
2. L.L. Peterson and B.S. Davie, “Computer Networks: A Systems Approach”, 2nd Ed., Morgan Kaufman, Harcourt Asia, 2000.
3. W. Stallings, “Data and Computer Communications”, 6th Ed., Pearson Education Asia (LPE), 2000.
4. F. Halsall, “Data Communications, Computer Networks and Open Systems”, 4th Ed., Pearson Education Asia (LPE), 1996.
5. L. Garcia and I. Widjaja, “Communication Networks: Fundamental Concepts and Key architectures”, Tata-McGraw-Hill Ed., 2000.
6. J.F. Kurose and K.W. Ross, “Computer Networking: A Top-Down Approach Featuring the Internet”, Pearson Education Asia (LPE), 2001.
7. L. Kleinrock, “Queueing Systems, Vol. 1: Theory”, John Wiley, 1975.

Database Management Systems Laboratory	**MCA0303C**
0 - 0 - 4 : 2 Credits	Prerequisites: *None*

Course Outcomes

At the end of the Course student will be introduced to

- Design and implement a database schema for a given problem-domain.
- Create and maintain tables using PL/SQL Course Outcome
- Populate and query a database, Prepare reports
- Application development using PL/SQL & front end tools

Syllabus

Exercises will be based on the syllabus of subject '**Database Management Systems**'.

Computer Network Laboratory	**MCA0304C**
0 - 0 - 4 : 2 Credits	Prerequisites: *None*

Course Outcomes

At the end of the Course student will be introduced to

- Understand computer network basics, network architecture, TCP/IP and OSI reference models; Identify and understand various techniques and modes of transmission
- Describe data link protocols, multi-channel access protocols and IEEE 802 standards for LAN
- Describe routing and congestion in network layer with routing algorithms and classify IPV4 addressing scheme
- Discuss the elements and protocols of transport layer; Understand network security and define various protocols such as FTP, HTTP, Telnet, DNS

Syllabus

Exercises will be based on the syllabus of subject '**Data Communication & Computer Network**'.

Project Phase I	**MCA0305C**
0 - 0 - 4 : 2 Credits	Prerequisites: *None*

Course Outcomes

At the end of the Course student will be introduced to

- Collaborate to learn new content and gain diverse perspectives.
- Understand the concept of any particular hardware software area and survey it.
- Acquire skills like collaboration, communication and independent learning, lifelong learning and the challenges ahead.
- Define a problem and try to conceptualize a solution

Syllabus

This shall be the initial phase of project.

Forth Semester

Project and Viva Voce	**MCA0401C**
0 - 0 - 20 : 10 Credits	Prerequisites: *None*

Course Outcomes

At the end of the Course student will be introduced to

- Acquire the ability to make links across different areas of knowledge and to generate, develop and evaluate ideas and information so as to apply these skills to the project task.
- Achieve skills to communicate effectively and to present ideas clearly and coherently to specific audience in both the written and oral forms.
- Attain collaborative skills through working in a team to achieve common goals.
- Learn on their own, reflect on their learning and take appropriate actions to improve it.

Syllabus

This shall be the final phase of project work.

Elective Subjects

Departmental Electives (16 Credits) to be completed in 2nd year (Semester III & IV)

Adhoc & Sensor Networks	**MCA0001E**
4 - 0 - 0 : 4 Credits	Prerequisites: *Basic concepts on Data Communications and Networking*

Course Outcomes

At the end of the Course student will be introduced to

- Understanding of various aspects of Adhoc sensor networks, describe the concepts, implementation, and usage.
- Discuss the challenges in designing MAC, routing and transport.
- Describe protocols for wireless ad-hoc/sensor networks
- Describe and implement protocols on a sensor testbed network and propose, implement, and evaluate new ideas for solving wireless sensor network design issues.

Syllabus

MANET - Introduction, Self-organizing behavior, Co-operation, MAC, Routing;
Multicast routing, Mobility model, Transport layer,
Opportunistic Mobile Networks, UAV networks, Wireless Sensor;
Networks (Introduction)- WSN (Coverage, Topology management), Mobile Sensor Networks;
MAC, Congestion control, Routing; Underwater WSN;
Security, Structure of sensor nodes;

Advanced Networking	**MCA0002E**
4 - 0 - 0 : 4 Credits	Prerequisites: *Computer Networks*

Course Outcomes

At the end of the Course student will be introduced to

- Differentiate between different LAN-based forwarding devices so that they can make thoughtful suggestions on how to build a network.
- Write networking code that uses TCP and UDP in client-server applications.
- Design and implement networking protocols.
- Design and implement networking applications.

Syllabus

Review for networking basics and IP networks, Introduction to wireless networks
Introduction of Algorithm Design and Optimization and their applications in networking. Scheduling algorithms and MAC layer protocols;
Routing algorithms and protocols (network layer), Congestion control algorithms and protocols
Cross-layer design, Quality of Service (QoS) provisioning; Network security

References:

1.Computer Networks: A Systems Approach (4th Edition) by Larry Peterson and Bruce Davie. Morgan Kaufmann.

Advances in Database	**MCA0003E**
4 - 0 - 0 : 4 Credits	Prerequisites: *None*

Course Outcomes

At the end of the Course student will be introduced to

- To provide a strong foundation in advanced database concepts from an industry perspective.
- To covers advanced data modeling concepts like OOD Modeling and ORD Modeling
- To learn query processing and transaction management concepts for object-relational database and distributed database.

Syllabus

Data Storage and Querying : Storage and File Structure - Indexing and Hashing –Physical Database Design and Tuning - Query Processing Algorithms – Query Optimization Techniques – Transaction Management: Transaction Processing Concepts – Concurrency Control – Recovery Techniques – Database Security.

Database System Architectures: Centralized and Client-Server Architectures – Server System Architectures – Parallel Systems- Distributed Systems – Parallel Databases: I/O Parallelism – Inter and Intra Query Parallelism – Inter and Intra operation Parallelism –Distributed Database Concepts - Distributed Data Storage – Distributed Transactions – Commit Protocols – Concurrency Control – Distributed Query Processing – Three Tier Client Server Architecture- Case Studies.

Concepts for Object Databases: Object Identity – Object structure – Type Constructors – Encapsulation of Operations – Methods – Persistence – Type and Class Hierarchies –Inheritance – Complex Objects – Object Database Standards, Languages and Design: ODMG Model – ODL – OQL – Object Relational and Extended – Relational Systems : Object Relational features in SQL / Oracle – Case Studies.

Active Database Concepts and Triggers – Temporal Databases – Spatial Databases – Multimedia Databases – Deductive Databases – XML Databases: XML Data Model – DTD - XML Schema - XML Querying - Geographic Information Systems - Genome Data Management.

Mobile Databases: Location and Handoff Management - Effect of Mobility on Data Management - Location Dependent Data Distribution - Mobile Transaction Models - Concurrency Control - Transaction Commit Protocols – Web Databases – Information Retrieval - Data Warehousing - Data Mining.

References:

1. R. Elmasri, S.B. Navathe, “Fundamentals of Database Systems”, Fifth Edition, Pearson Education/Addison Wesley, 2007.
2. Thomas Cannolly and Carolyn Begg, “ Database Systems, A Practical Approach to Design, Implementation and Management”, Third Edition, Pearson Education, 2007.
3. Henry F Korth, Abraham Silberschatz, S. Sudharshan, “Database System Concepts”, Fifth Edition, McGraw Hill, 2006.
4. C.J.Date, A.Kannan and S.Swamynathan,”An Introduction to Database Systems”, Eighth Edition, Pearson Education, 2006.

Raghu Ramakrishnan, Johannes Gehrke, “Database Management Systems”, McGraw Hill, Third Edition 2004.

Artificial Intelligence	**MCA0004E**

4 - 0 - 0 : 4 Credits Prerequisites: *None*

Course Outcomes

At the end of the Course student will be introduced to

- Identify the type of an AI problem (search, inference, decision making under uncertainty, game theory, etc).
- Formulate the problem as a particular type. (Example: define a state space for a search problem)
- Compare the difficulty of different versions of AI problems, in terms of computational complexity and the efficiency of existing algorithms.
- Implement, evaluate and compare the performance of various AI algorithms. Evaluation could include empirical demonstration or theoretical proofs.

Syllabus

Introduction to AI. The foundations of AI. Importance of AI and related fields; Propositional and predicate logic, representation atoms, connectives, literals, CNF, DNF and casual form, interpretation and model, satisfiability, resolution principle and unification; **Rules:** working memory, rule base, conflict set, conflict resolution strategies, backward and foreword chaining, meta rules.

Reasoning under Uncertainty: basic probability notation,probabilistic reasoning, Bayesian networks, certainty factor methods, Dempster-Shafer theory, basics of fuzzy logic; Structure Representation**:** semantic networks, frames, conceptual dependency, scripts, inheritance, default values**.**

Logical agents, reasoning and resolution, adequacy, richness, granularity, ease of representation and use, modeling uncertainty, the fame problem, declarative and procedural representation; Problem solving by Searching: State space repetition, heuristics, heuristic evolution function, and problem reduction. Searching for solutions. Informed and uninformed search strategies.

Search Methods**:** generate and test, hill climbing, means-ends analysis, depth-first, breath-first, best first, exploiting domain constraints, dependency-directed back tracking, minimax, alpha- beta pruning, iterative deepening; Planning by forward and backward reasoning, Nonlinear planning, scheduling

References:

1. Artificial Intelligence by E Rich and K Knight, KcGraw-Hill.
2. Artificial Intelligence (3rd Ed) PH Winston, Addison-Wesley.
3. Introduction of Artificial Intelligence and expert systems by DW Patterson, PHI.
4. Artificial Intelligence a Modern Approach-Stuart Russell, Peter Norvig, PHI
5. Artificial Intelligence and Soft Computing by A. Konar, CRC Press 2000

Cloud Computing MCA0005E

4 - 0 - 0 : 4 Credits Prerequisites: *None*

Course Outcomes

At the end of the Course student will be introduced to

- Introduction to Cloud Computing and Architecture
- Understanding Service, Data and Resource Management in Cloud Computing
- Improvising Cloud Security, Open Source and Commercial Clouds, Cloud Simulator
- Ideating research trends in Cloud Computing and Fog Computing

Syllabus

Cloud architecture and model: Technologies for Network-Based System – System Models for Distributed and Cloud Computing – NIST Cloud Computing Reference Architecture. Cloud Models:- Characteristics – Cloud Services – Cloud models (IaaS, PaaS, SaaS) – Public vs Private Cloud –Cloud Solutions - Cloud ecosystem – Service management – Computing on demand.

Basics of Virtualization - Types of Virtualization - Implementation Levels of Virtualization - Virtualization Structures - Tools and Mechanisms - Virtualization of CPU, Memory, I/O Devices - Virtual Clusters and Resource management – Virtualization for Data-center Automation.

Cloud infrastructure: Architectural Design of Compute and Storage Clouds – Layered Cloud Architecture Development – Design Challenges - Inter Cloud Resource Management – Resource Provisioning and Platform Deployment – Global Exchange of Cloud Resources.

Programming model: Parallel and Distributed Programming Paradigms – MapReduce , Twister and Iterative MapReduce – Hadoop Library from Apache – Mapping Applications - Programming Support - Google App Engine, Amazon AWS - Cloud Software Environments -Eucalyptus, Open Nebula, OpenStack, Aneka, CloudSim.

Security in the cloud: Security Overview – Cloud Security Challenges and Risks – Software-as-a-Service Security – Security Governance – Risk Management – Security Monitoring – Security Architecture Design – Data Security – Application Security – Virtual Machine Security - Identity Management and Access Control – Autonomic Security.

REFERENCES:

1. Cloud Computing: Principles and Paradigms, Editors: Rajkumar Buyya, James Broberg, Andrzej M. Goscinski, Wiley,2011
2. Enterprise Cloud Computing - Technology, Architecture, Applications, Gautam Shroff, Cambridge University Press, 2010
3. Cloud Computing Bible, Barrie Sosinsky, Wiley-India, 2010
4. Cloud Security: A Comprehensive Guide to Secure Cloud Computing, Ronald L. Krutz, Russell Dean Vines, Wiley- India,2010

Cryptography and Network Security	**MCA0006E**
4 - 0 - 0 : 4 Credits	Prerequisites: *Computer Networks*

Course Outcomes

At the end of the Course student will be introduced to

- Analyze and design classical encryption techniques and block ciphers.
- Understand and analyze data encryption standard.
- Understand and analyze public-key cryptography, RSA and other
- Public-key cryptosystems such as Diffie-Hellman Key Exchange, ElGamal Cryptosystem.

Syllabus

Introduction to cryptography, Classical Cryptosystem, Cryptanalysis on Substitution Cipher, Play fair Cipher, Block Cipher; Data Encryption Standard (DES), Triple DES, Modes of Operation, Stream Cipher, Pseudorandom Sequence;

LFSR based Stream Cipher, Mathematical background, Abstract algebra, Number Theory;

Modular Inverse, Extended Euclid Algorithm, Fermat's Little Theorem, Eular Phi-Function, Eular's theorem, Quadratic Residue, Polynomial Arithmetic; Advanced Encryption Standard (AES), Introduction to Public Key Cryptosystem, Diffie-Hellman Key Exchange, Knapsack Cryptosystem, RSA Cryptosystem.

More on RSA, Primarily Testing, ElGamal Cryptosystem, Elliptic Curve over the Reals, Elliptic curve Modulo a Prime; Generalised ElGamal Public Key Cryptosystem, Chinese Remainder Theorem, Rabin Cryptosystem, Legendre and Jacobi Symbol.

Message Authentication, Digital Signature, Key Management, Key Exchange, Hash Function; Universal Hashing, Cryptographic Hash Function, Secure Hash Algorithm (SHA), Digital Signature Standard (DSS), More on Key Exchange Protocol.

Cryptanalysis, Time-Memory Trade-off Attack, Differential Cryptanalysis, More on Differential Cryptanalysis, Linear Cryptanalysis; Cryptanalysis on Stream Cipher, Algebraic Attack, Implementation Attacks, side channel attack.

Internetwork Security, SSL, PGP, Cloud Security, Introduction to Blockchain and Bitcoin.

Data Mining	**MCA0007E**
4 - 0 - 0 : 4 Credits	Prerequisites: *None*

Course Outcomes

At the end of the Course student will be introduced to

- Find a meaningful pattern in data.
- Graphically interpret data.
- Implement the analytic algorithms.
- Handle large scale analytics projects from various domains.

Syllabus

Introduction, Data Preprocessing
Association Rule Mining, Classification Basics
Decision Tree, Bayes Classifier, K nearest neighbor
Support Vector Machine, Kernel Machine
Clustering, Outlier detection
Sequence mining
Evaluation, Visualization.

Data Science	**MCA0008E**
4 - 0 - 0 : 4 Credits	Prerequisites: *None*

Course Outcomes

At the end of the Course student will be introduced to

- Proficiency with statistical analysis of data.
- Ability to build and assess data-based models.
- Execute statistical analyses with professional statistical software. Skill in data management.
- Apply data science concepts and methods to solve problems in real-world contexts and will communicate these solutions effectively issues.

Syllabus

Course philosophy and introduction to R; Linear algebra for data science: Algebraic view - vectors, matrices, product of matrix & vector, rank, null space, solution of over-determined set of equations and pseudo-inverse) ; Geometric view - vectors, distance, projections, Eigen value decomposition;

Statistics (descriptive statistics, notion of probability, distributions, mean, variance, covariance, covariance matrix, understanding univariate and multivariate normal distributions, introduction to hypothesis testing, confidence interval for estimates)

Optimization; Typology of data science problems and a solution framework;

Simple linear regression and verifying assumptions used in linear regression; Multivariate linear regression, model assessment, assessing importance of different variables, subset selection;

Classification using logistic regression; Classification using knn and k-means clustering

Suggested reading materials:

1. Introduction to linear algebra - by gilbert strang
2. Applied statistics and probability for engineers – by douglas montgomery

Deep Learning	**MCA0009E**
4 - 0 - 0 : 4 Credits	Prerequisites: *Linear Algebra, Probability Theory*

Course Outcomes

At the end of the Course student will be introduced to

- Know the main techniques in deep learning and the main research in this field.
- Able to identify new application requirements in the field of computer vision.
- Able to identify reasonable work goals and estimate the resources required to achieve the objectives.
- Able to structure and prepare scientific and technical documentation describing project activities.

Syllabus

History of Deep Learning, Deep Learning Success Stories, McCulloch Pitts Neuron, Thresholding Logic, Perceptrons, Perceptron Learning Algorithm
Multilayer Perceptrons (MLPs), Representation Power of MLPs, Sigmoid Neurons, Gradient Descent, Feedforward Neural Networks, Representation Power of Feedforward Neural Networks
FeedForward Neural Networks, Backpropagation
Gradient Descent (GD), Momentum Based GD, Nesterov Accelerated GD, Stochastic GD, AdaGrad, RMSProp, Adam, Eigenvalues and eigenvectors, Eigenvalue Decomposition, Basis
Principal Component Analysis and its interpretations, Singular Value Decomposition
Autoencoders and relation to PCA, Regularization in autoencoders, Denoising autoencoders, Sparse autoencoders, Contractive autoencoders
Regularization: Bias Variance Tradeoff, L2 regularization, Early stopping, Dataset augmentation, Parameter sharing and tying, Injecting noise at input, Ensemble methods, Dropout
Greedy Layerwise Pre-training, Better activation functions, Better weight initialization methods, Batch Normalization
Learning Vectorial Representations Of Words
Convolutional Neural Networks, LeNet, AlexNet, ZF-Net, VGGNet, GoogLeNet, ResNet, Visualizing Convolutional Neural Networks, Guided Back propagation, Deep Dream, Deep Art, Fooling Convolutional Neural Networks
Recurrent Neural Networks, Back propagation through time (BPTT), Vanishing and Exploding Gradients, Truncated BPTT, GRU, LSTMs
Encoder Decoder Models, Attention Mechanism, Attention over images

REFERENCES:

1. Deep Learning, An MIT Press book, Ian Goodfellow and Yoshua Bengio and Aaron

Digital Logic & Basic Electronics	**MCA0010E**
4 - 0 - 0 : 4 Credits	Prerequisites: *None*

Course Outcomes

At the end of the Course student will be introduced to

- Design and analyse sequential logic circuits and synchronous finite state machines.
- To learn the Basics of HDL modelling and design techniques.
- To Design controller using ASM chart method.
- Design and analysis of asynchronous finite state machines.

Syllabus

Digital system and binary numbers: Signed binary numbers, binary codes, cyclic codes parity generators and checkers, hamming codes. Floating point representation; Gate-level minimization: The map method up to five variable, don't care conditions, POS simplification, NAND and NOR implementation.

Combinational Logic: Combinational circuits, analysis procedure, design procedure, binary adder-subtractor, decimal adder, binary multiplier, magnitude comparator, decoders, encoders, multiplexers, demultiplexers;

Sequential logic: Sequential circuits, storage elements: latches, flip flops, analysis of clocked sequential circuits, state reduction and assignments, design procedure; Registers and counters: Shift registers, ripple counter, synchronous counter, other counters.

Memory Devices: Metal-Oxide-Semiconductor Field Effect Transistor (MOS), Complementary Metal-Oxide-Semiconductor Field Effect ransistor (CMOS), Charge- Coupled Device (CCD).

Network Theorems: Statements of Thevemin's, Norton's-, Superposition- and Maximum Power Transfer Theorems and their uses; Diode: Forward Bias and Reverse Bias: Use as Rectifier (Half Wave, Full Wave and Bridge); Load Regulation and Line regulation; Transistors: Common Base, Common Emitter and Common Collector Configurations, Transistor a and b , Their Relation. Common Emitter Amplifier; Power Supply: Circuit with 78** and 79** series, SMPS

Text Books:

1. Morris Mano, "Digital logic and Computer Design", Prentice-Hall of India, 1998.
2. Integrated Electronics: Milman and Halkias

References:

1. Digital Principles and applications: Malvino and Leach
2. Digital Principles: Schaum's Series
3. William I. Fletcher, "An Engineering Approach to Digital Design ", PHI, 1980.
4. Floyd T.L.,"Digital Fundamentals", Charles E. Merrill publishing Company, 1982.
5. TokheimR.L.,"Digital Electronics- Principles and Applications ", Tata McGraw Hill, 1999.
6. Jain R.P., "Modern Digital Electronics ", Tata McGraw Hill, 1999.

Digital Signal Processing	**MCA0011E**
4 - 0 - 0 : 4 Credits	Prerequisites: B*ackground of signals and systems*

Course Outcomes

At the end of the Course student will be introduced to

- Understand signal processing systems using basic concepts.
- Analyze signal using the discrete Fourier transform and its effective computation by FFT techniques.
- Specify and design FIR and IIR type digital filters and identify the fundamentals of multi rate signal processing and its applications.
- Understand advanced digital signal processing techniques.

Syllabus

Properties of a discrete-time linear time-invariant system, Compare the properties of different signal types including: continuous-time signal, discrete-time signals, periodic signals, a periodic signals, deterministic signals, and random signals, Calculate the power and energy of a signal, properties of fundamental signals using mathematical equations which includes impulse function, unit-step function, sinusoid function, and exponential function, Apply Euler's formula to represent exponentials in terms of sinusoidal signals, function, and properties of convolution, application of complex exponential terms to calculate the Fourier Transform and Inverse Fourier Transform of a signal, application of the Fourier Transform to represent time and frequency domains of an impulse, a constant, a rectangular pulse, a sinc function, a sinusoid, and an impulse train, demonstration of the scaling, frequency shifting, and convolution properties of the Fourier Transform.

Identify and compare signal behavior during sampling and reconstruction in frequency and time domains, the implication of low pass filtering in frequency and time domains, convolution properties of the Fourier Transform during signal sampling, the effects of signal aliasing, the importance and implementation of antialiasing filters and the Nyquist theorem in signal processing applications, define the z-transform in the representation of signals and systems, the usage of z-transform and its relationship with Laplace transform, the equivalences between z-transform and DTFT (Discrete-Time Fourier Transform), the properties of z-transform zeros, poles, and Region of Convergence (ROC), demonstration of the linearity, shift, convolution, and order of summation properties of z-transform, the convolution and multiplication properties between time domain and z-domain.

Properties of a moving average filter, the mathematical properties of the impulse and frequency response of a Finite Impulse Response (FIR) filter and a moving average filter, working of FIR filter, relationship between an FIR filter and a moving average filter in terms of filter coefficients, relationship between the continuous frequency response of an FIR filter and the DTFT of the filter coefficients, mathematical and graphical form the frequency response of low pass, high pass, bandpass, and bandstop filters, designing of an FIR filter using the window method, the mathematical form for an Infinite Impulse Response (IIR) filter, the properties of impulse invariance method for IIR filter design, the properties of the bilinear transform method for IIR filter design, the implementation of different structure types of IIR filter including Direct form I, Direct form II, and cascaded Second-Order Stage (SOS).

Usage of the Fourier Transform, Fourier Series, Discrete-Time Fourier Transform (DTFT) and Discrete Fourier Transform (DFT) for various signal types in time and frequency domains, Fast Fourier Transform (FFT) and explain its relationship with DFT, properties of twiddle factors Fast Fourier Transform(FFT) in mathematical form using twiddle factors, calculation of Discrete Fourier Transform (DFT), the relationship between N-point DFT with N/2-point DFT of even and odd values of the signal, the benefits of the radix method for FFT and its computational savings, characteristics of adaptive systems, functionality and operation of a closed-loop configuration involving adaptive filters, applications for the system identification configuration with adaptive filters, the concept and purpose of the Steepest Descent and the Least Means Squares (LMS) algorithm.

SUGGESTED READING

1. "Digital Signal Processing" by A. Oppenheim and R. Schafer
2. "Discrete Time Signal Processing" by A. Oppenheim and R. Schafer
3. "Digital Signal Processing" by J. G. Proakis and D. G. Manolakis
4. "Digital Signal Processing" by S. K. Mitra
5. "Digital Signal Processing using Arm Cortex-M based Microcontrollers: Theory and Practice" https://www.arm.com/resources/education/textbooks /dsptextbook
6. "Digital Signal Processing Using the ARM Cortex M4 Paperback" by Donald S. Reay

Discrete Mathematical Structures	**MCA0012E**
4 - 0 - 0 : 4 Credits	Prerequisites: *none*

Course Outcomes

At the end of the Course student will be introduced to

- Introduce concepts of mathematical logic for analyzing propositions and proving theorems.
- Use sets for solving applied problems, and use the properties of set operations algebraically.
- Work with relations and investigate their properties AND Investigate functions as relations and their properties.
- Introduce basic concepts of graphs, digraphs and trees.

Syllabus

Sets, relations, and functions:- Basic operations on sets, Cartesian products, disjoint union and power sets. Types of relations, compositions and inverses; Types of functions, compositions and inverses; Arbitrary union, intersection and product; Propositional Logic:- Proof systems, Satisfiability, Validity, Soundness, Completeness, Deduction Theorem. Decision problems on propositional logic;

First order logic and First order theory, Set theory, Axiom of choice. Finite and Infinite sets, Countable and Uncountable, Cantor's diagonal argument and power set theorem, non-computability of all number theoretic; functions.

Partially ordered sets: Complete partial ordering, chain, lattice. Complete, distributive, modular, and complemented lattices. Boolean and pseudo Boolean lattices. Different sublattices, monotone map and morphisms, quotient structures, filters. Tarski's fixed point's theorem.

Algebraic Structures: Algebraic structures with one binary operation, semigroup, monoid and group. Congruence relation and quotient structures. Morphisms. Free and cyclic monoids and groups. Permutation group. Substructures, normal subgroup. Error correcting code. Algebraic structures with two binary operations- ring, integral domain, and field. Boolean algebra and Boolean ring.

Counting: Basic counting techniques, inclusion and exclusion, pigeon-hole principle, permutation, Combination, Summations. Recurrence Relation and Generating Function.

Graph: Graphs and their basic properties, Eulerian and Hamiltonian walk, Graph Colouring, Trees.

Text Books:

1. Liptschutz, Seymour, " Discrete Mathematics", McGraw Hill.
2. Tremblay and Sorenson, Discrete Mathematical Structures, McGraw-Hill.

References:

1. C L Lieu, "Elements of Discrete Mathematics"
2. B. Kolman, R. Busby, R. C. Ross, "Discrete Mathematics", Pearson

3. R. L. Graham, D. E. Knuth, and O. Patashnik, Concrete Mathematics, 2nd Ed, Addison-Wesley, 1994.

Distributed Computing	**MCA0013E**
4 - 0 - 0 : 4 Credits	Prerequisites: *none*

Course Outcomes

At the end of the Course student will be introduced to

- Demonstrate knowledge of the basic elements and concepts related to distributed system technologies.
- Demonstrate knowledge of the core architectural aspects of distributed systems. Design and implement distributed applications.
- Use and apply important methods in distributed systems to support scalability and fault tolerance.
- Design and implement distributed applications and Demonstrate experience in building large-scale distributed applications.

Syllabus

Computer architecture: CICS, RISC, Multi-core; Computer networking : ISO/OSI Model; Evolution of operating systems; Introduction to distributed computing systems (DCS) DCS design goals, Transparencies, Fundamental issues

Distributed Coordination: Temporal ordering of events; Lamport's logical clocks; Vector clocks; Ordering of messages; Physical clocks; Global state detection

Process synchronization: Distributed mutual exclusion; Algorithms; Performance matrix

Inter-process communication: Message passing communication; Remote procedure call; Transaction communication; Group communication; Broadcast atomic protocols; Deadlocks in distributed systems, Load scheduling and balancing techniques, Distributed database system: A Case study

Text Books:

1. Distributed and Cloud Computing: Clusters, Grids, Clouds, and the Future Internet (DCC) by Kai Hwang, Jack Dongarra& Geoffrey C. Fox.
2. "Distributed Systems: Principles and Paradigms" (DSPD), Prentice Hall, 2nd Edition, 2007.

Formal Language and Automata Theory	**MCA0014E**
4 - 0 - 0 : 4 Credits	Prerequisites: *none*

Course Outcomes

At the end of the Course student will be introduced to

- Outline of different concepts in automata theory and formal languages.
- Structure of Regular Languages (RL), Regular Expressions (RE) and construction of corresponding Finite Automata (FA).
- Study of Context Free Grammar (CFG) and designing Push Down Automata (PDA).
- Explain and designing of Turing Machines (TM). Concept of un-decidable problems related to TM and CFG.

Syllabus

Finite Automata – deterministic and nondeterministic, regular operations; Regular Expression, Equivalence of DFA, NFA and REs, closure properties; Non regular languages and pumping lemma, DFA Minimization,
CFGs, Chomsky Normal Form; Non CFLs and pumping lemma for CFLs, PDAs, Equivalence of PDA and CFG; Properties of CFLs, DCFLs, Turing Machines and its variants
Configuration graph, closure properties of decidable languages, decidability properties of regular languages and CFLs;
Undecidability, reductions, Rice's Theorem, introduction to complexity theory

SUGGESTED READING

1. Introduction to the Theory of Computation by Michael Sipser.

- Identify and analyze the fundamental steps in Image processing.
- Characterize the hardware and software components of imaging systems.
- Understand the models and interpret the spatial and frequency domain image processing algorithms and analyze and verify different image recognition techniques.
- Apply the concepts and image processing tools for different image processing and pattern recognition applications.

Image Processing	**MCA0015E**
4 - 0 - 0 : 4 Credits	Prerequisites: *none*

Course Outcomes

At the end of the Course student will be introduced to

- Identify and analyze the fundamental steps in Image processing.
- Characterize the hardware and software components of imaging systems.
- Understand the models and interpret the spatial and frequency domain image processing algorithms and analyze and verify different image recognition techniques.
- Apply the concepts and image processing tools for different image processing and pattern recognition applications.

Syllabus

Introduction and signal digitization; Pixel relationship; Camera models & imaging geometry
Image interpolation; Image transformation; Image enhancement
Image restoration & Image registration
Colour image processing; Image segmentation
Morphological image processing; Object representation, description and recognition

SUGGESTED READING

1. Digital Image Processing by Rafael C Gonzalez & Richard E Woods, 3rd Edition
2. Fundamentals of Digital Image Processing by Anil K Jain
3. Digital Image Processing by William K Pratt

Information Retrieval and Web Mining	**MCA0016E**
4 - 0 - 0 : 4 Credits	Prerequisites: *None*

Course Outcomes

At the end of the Course student will be introduced to

- Understand the basic concepts of the information retrieval.
- Analyse the involvement of the information retrieval in modern life style & social media.
- Apply data pre-processing, indexing, retrieval methods and concepts
- Evaluate the effectiveness and efficiency of different information retrieval systems.

Syllabus

Information Retrieval: The nature of unstructured and semi-structured text. Inverted index and Boolean queries. Text Indexing, Storage and Compression: Text encoding: tokenization, stemming, stop words, phrases, index optimization. Index compression: lexicon compression and postings lists compression. Gap encoding, gamma codes, Zipf's Law. Index construction. Postings size estimation, merge sort, dynamic indexing, positional indexes, n-gram indexes, real-world issues.

Retrieval Models: Boolean, vector space, TFIDF, Okapi, probabilistic, language modeling, latent semantic indexing. Vector space scoring. The cosine measure. Efficiency considerations. Document length normalization. Relevance feedback and query expansion. Rocchio; Performance Evaluation: Evaluating search engines. User happiness, precision, recall, F-measure. Creating test collections: kappa measure, interjudge agreement.

Text Categorization and Filtering: Introduction to text classification. Naive Bayes models. Spam filtering. Vector space classification using hyperplanes; centroids; k Nearest Neighbors. Support vector machine classifiers. Kernel functions. Boosting; Text Clustering: Clustering versus classification. Partitioning methods. k-means clustering. Mixture of gaussians model. Hierarchical agglomerative clustering. Clustering terms using documents.

Web Information Retrieval: Hypertext, web crawling, search engines, ranking, link analysis, PageRank, HITS; Retrieving Structured Documents: XML retrieval, semantic web; Advanced Topics: Summarization, Topic detection and tracking, Personalization, Question answering, Cross language information retrieval

References:

1. Introduction to Information Retrieval, Manning, Raghavan and Schutze, Cambridge University Press, draft.
2. Modern Information Retrieval Baeza-Yates and Ribeiro-Neto, Addison Wesley, 1999
3. Mining the Web, SoumenCharabarti, Morgan-Kaufmann, 2002.

Internet of Things	**MCA0017E**
4 - 0 - 0 : 4 Credits	Prerequisites: *Basic programming knowledge*

Course Outcomes

At the end of the Course student will be introduced to

- Understand advanced networking concepts and internet and web application architectures.
- Analyze and understand different advanced routing protocols being used in web application development.
- Analyze and evaluate different solution available in the field of networking and web application development
- Implement solution for different critical network related issue.

Syllabus

Introduction to IoT, What is IoT, Why and How, Challenges, Technologies that enable IoT, Mobile Devices – ARM Cortex-A class processorIntroduction to Embedded Systems, CPUs vs. MCUs vs. Embedded Systems, , Embedded Devices – ARM Cortex-M class processor, Networking – Bluetooth Smart technology, Connected Community – ARM mbed platform, Features of Embedded Systems, Building Embedded Systems , Building Embedded System using MCUs, Introduction to the mbedTM Platform.

ARM Architectures and Processors, ARM Processors Families, ARM Cortex-M Series Family, Cortex-M0 Processor, ARM Processor Vs. ARM Architectures, ARM Cortex-M0 Processor, Cortex-M0 Processor Overview, Cortex-M0 Block Diagram, Cortex-M0 Registers, Cortex-M0 Memory Map, Cortex-M0 Exception Handling, ARM Cortex-M0 Processor Instruction Set, ARM and Thumb Instruction Set, Cortex-M0 Instruction Set, Data Accessing Instructions, Arithmetic Instructions, Program Flow Control, Low-Power Features, Low-Power Requirements, Cortex-M0 Low-Power Features, Cortex-M0 Sleep Mode, Developing Low-Power Applications.

Interrupts, Entering an Exception Handler, Exiting an Exception Handler, Port Module and External Interrupts, Timing Analysis, Program Design with Interrupt, Bluetooth overview, Bluetooth Key Versions, Bluetooth Low Energy (BLE) Protocol, Bluetooth, Low Energy Architecture, Introduction to Smartphone, Smartphone hardware components, Mobile operating systems and Smartphone programming.

PSoC Terms, The Internet of Things Revolution, Design Challenge 1, Design Challenge 2, Design Challenge 3, PSoC 4 BLE One-Chip Solution, Bluetooth Low Energy Pioneer Kit, BLE Pioneer Baseboard and PSoC 4 BLE Module, BLE architecture and Component Overview, Programmable Analog Blocks, Sequencing SAR ADC Block, Continuous Time Block(CTBm), Programmable Digital Blocks, Universal Digital Block, Serial Communication Block, Timer/Counter/PWM Block, Sensor-Based IoT System Design.

Practical Implementations: Blinking LED, CPU Project: Square root approximation, C as implemented in Assembly Lab Exercise. Interrupt and low POWER features, Introduction to the ARM DS-5 TOOL, C and assembly coding - Processing Text in Assembly Language, Designing a basic Mobile Appcesssory, Bluetooth Smart Heart Rate Sensor App, BLE connection and BLE heart rate monitor, Capsensor and RGB led

Text Books and references:

- "The Internet of Things: Enabling Technologies, Platforms, and Use Cases", by Pethuru Raj and Anupama C. Raman (CRC Press)
- "Internet of Things: A Hands-on Approach", by Arshdeep Bahga and Vijay Madisetti (Universities Press
- White Paper: Cortex-M for Beginners - An overview of the Arm Cortex-M processor family and comparison: https://community.arm.com/developer/ip-products/processors/b/processors-ip-blog/posts/white-paper-cortex-m-for-beginners-an-overview-of-the-arm-cortex-m-processor-family-and-comparison
- The Definitive Guide to ARM® Cortex®-M3 and Cortex®-M4 Processors, Third Edition by Joseph Yiu
- Cortex-A Series Programmer's Guide for ARMv7-A by Arm http://infocenter.arm.com/help/topic/com.arm.doc.den0013d/index.html

Internet Technology	**MCA0018E**
4 - 0 - 0 : 4 Credits	Prerequisites: *None*

Course Outcomes

At the end of the Course student will be introduced to

- Analyze a web page and identify its elements and attributes.
- Create web pages using XHTML and Cascading Style Sheets.
- Build dynamic web pages using JavaScript (Client side programming).
- Create XML documents and Schemas.
- Build interactive web applications using AJAX.

Syllabus

Web essentials – W3C - clients – servers - communication – markup languages – XHTML – simple XHTML pages style sheets – CSS; Client side programming – Java script language – java script objects – host objects Browsers and the DOM; Server side programming – Java servlets – basics – simple program – separating programming and presentation – ASP/JSP - JSP basics ASP/JSP objects – simple ASP/JSP pages.

Representing Web data – data base connectivity – JDBC – Dynamic Web pages – XML – DTD

XML schema – DOM – SAX – XQuery - Building Web applications - cookies – sessions – open source environment – PHP – MYSQL –case studies; Middleware Technologies – Ecommerce – Architectures – Technologies – Ajax – Advanced Web Technologies and Tools.

REFERENCES:

1. Jeffrey C Jackson, "Web Technology – A computer Science perspective", Persoson Education, 2007.
2. Chris Bates, "Web Programming – Building Internet Applications, "Wiley India, 2006.

Machine Learning	**MCA0019E**
4 - 0 - 0 : 4 Credits	Prerequisites: *Basic programming, algorithm design, basics of probability & statistics*

Course Outcomes

At the end of the Course student will be introduced to

- Understand the basic theory underlying machine learning.
- Able to formulate machine learning problems corresponding to different applications.
- Understand a range of machine learning algorithms along with their strengths and weaknesses. Learn various AI paradigms.
- To be able to apply machine learning algorithms to solve problems of moderate complexity. Learn various ML tools.

Syllabus

Introduction: Basic definitions, types of learning, hypothesis space and inductive bias, evaluation, cross-validation; Linear regression, Decision trees, overfitting
Instance based learning, Feature reduction, Collaborative filtering based recommendation
Probability and Bayes learning;
Logistic Regression, Support Vector Machine, Kernel function and Kernel SVM; Neural network: Perceptron, multilayer network, backpropagation, introduction to deep neural network
Computational learning theory, PAC learning model, Sample complexity, VC Dimension, Ensemble learning; Clustering: k-means, adaptive hierarchical clustering, Gaussian mixture model

REFERENCES:

1. Machine Learning. Tom Mitchell. First Edition, McGraw- Hill, 1997.
2. Introduction to Machine Learning Edition 2, by Ethem Alpaydin

Multimedia Technology	**MCA0020E**
4 - 0 - 0 : 4 Credits	Prerequisites: *None*

Course Outcomes

At the end of the Course student will be introduced to

- Understanding the basics of analog and digital video: video representation and transmission and analyze analog and digital video signals and systems.
- Knowing the fundamental video processing techniques.
- Acquiring the basic skill of designing video compression and to familiarize with video compression standards.
- Knowing the basic techniques in designing video transmission systems: error control and rate control.

Syllabus

Multimedia Technology-Introduction, Nature of Multimedia Data, Multimedia Peripherals & Devices, Storage of Multimedia Data; Multimedia systems design -An Introduction – Multimedia applications – Multimedia System Architecture – Evolving technologies for Multimedia – Defining objects for Multimedia systems – Multimedia Data interface standards – Multimedia Databases.

Different Data Compression Techniques: Run length encoding, quantization, wavelet transform, JPEG 2000 image compression, vector quantization and codebook method a Temporal Model for Interactive Multimedia; Multimedia file handling - Compression & Decompression – Data & File Format standards – Multimedia I/O technologies - Digital voice and audio – video image and animation – Full motion video – Storage and retrieval Technologies.

Multimedia Databases, Clustering for Multimedia Object Storage, Clustering Algorithms, Querying and Content Retrieval in Multimedia Databases, Distributed Multimedia Systems; Hypermedia - Multimedia Authoring & User Interface – Hypermedia messaging - Mobile Messaging – Hypermedia message component – creating Hypermedia message – Integrated multimedia message standards – Integrated Document management

References:

1. J. Keyes: Multimedia Handbook, MH.
2. G. Blair, L. Blair, A. Chetwynd, H. Bowman: Formal Specification of Distributed Multimedia Systems, UCL Press, London.
3. S. Khoshafian, A. Brad Baker: Multimedia and Imaging Databases, Morgan Kaufmann.

Natural Language Processing	**MCA0021E**
4 - 0 - 0 : 4 Credits	Prerequisites: *Basic probabilities knowledge*

Course Outcomes

At the end of the Course student will be introduced to

- Recognize the feasibility of applying a soft computing methodology for a particular problem.
- Apply fuzzy logic and reasoning to handle uncertainty and solve engineering problems.
- Apply genetic algorithm to combinational optimization problems.
- Apply neural networks to pattern classifications and regression problems. And compare solutions by various soft computing approaches for a given problem.

Syllabus

Introduction and Basic Text Processing, Spelling Correction, Language Modeling
Advanced smoothing for language modeling, POS tagging
Models for Sequential tagging – MaxEnt, CRF
Syntax – Constituency Parsing, Dependency Parsing
Lexical Semantics, Distributional Semantics
Topic Models , Entity Linking, Information Extraction
Text Summarization, Text Classification
Sentiment Analysis and Opinion Mining

Network Synthesis	**MCA0022E**
4 - 0 - 0 : 4 Credits	Prerequisites: *None*

Course Outcomes

At the end of the Course student will be introduced to

- Apply network topology concepts in the formulation and solution of electric network problems.
- Apply two-port network analysis in the design and analysis of filter and attenuator networks.
- Identify the properties and characteristics of network functions, and verify the mathematical constraints for their physical realisation.
- Synthesize passive one-port networks.

Syllabus

Network Topology: Graph of a network, concept of a tree and links, incidence matrix, principle of power inverience ,Tellegen's Theorem, Network transformations, principle of duality in network.
Transient behavior and initial conditions in network: Behavior of circuit elements under transient condition, transient response of a circuit for A.C. and D.C. excitations, Importance of initial condition, Evaluation of initial conditions in R-L, R-C and R-L-C circuits.

Laplace Transform: Introduction, definition of Laplace Transform pairs, Properties of Laplace Transform, Laplace Transform of some important functions, Laplace Transform theorem, Waveform synthesis.
Analysis of Networking Laplace Transform: Laplace Transform of periodic function and related problems. Sinusoidal functions and its related problems, Initial value and final value theorem, convolution theorem with problems;

Network Functions: Introduction, Network functions of single point network and two point network, Classification of network functions: Driving point, impedance function, driving point admittance function, Transfer function, Poles and Zeros;necessary conditions for transfer functions, related problems.

Network Synthesis: +ve real functions, Hurwitz polynomials, Elementary synthesis operations, properties of L-C, R-C and R-L impedances and admittance, Synthesis of of LL, RC and RL network using Foster and Cover methods.

Two point network Parameters: Open circuit impedance parameters, sort circuit admittance parameters. Transmission parameters. Hybrid Parameters. Relation between parameter sets. Calculations of these parameters for a given network.
Filters: General theory of filters, designing of filters, Prototype low pass, and high pass filters. Composite filters, Band pass and Band elimination filters.

Text books:

1. Network Theory: Analysis And Synthesis By SmarajitGhosh, Phi
2. Network Theory: A Simplified Approach By K ChannaVenkatesh& D Ganesh Rao,Pearson

References:

1. A First Course In Network Theory By Ernesto Estrada And Philip A. Knight , Oxford

Numerical Methods	**MCA0023E**
4 - 0 - 0 : 4 Credits	Prerequisites: *None*

Course Outcomes

At the end of the Course student will be introduced to

- Apply numerical methods to find solution of algebraic equations using different methods under different conditions, and numerical solution of system of algebraic equations.
- Apply various Differential Equations concepts and Multistep Methods.
- Derive numerical methods for various mathematical operations like interpolation, differentiation, integration, the solution of linear and nonlinear equations, and the solution.
- Work numerically on the partial differential equations using different methods through the theory of finite differences.

Syllabus

Solution of equations and Eigen value problems: Iterative method, Newton – Raphson method for single variable and for simultaneous equations with two variables. Solutions of a linear system by Gaussian, Gauss-Jordan, Jacobi and Gauss – Seidel methods.Inverse of a matrix by Gauss – Jordan method.
Eigen value of a matrix by Power and Jacobi methods.

Interpolation: Newton's divided difference formulae, Lagrange's and Hermite's polynomials. Newton forward and backward difference formulae. Stirling's and Bessel's Central difference formulae.

Numerical Differentiation And Integration: Numerical differentiation with interpolation polynomials, Numerical integration by Trapezoidal and Simpson's (both 1/3rd and 3/8th) rules. Two and Three point Gaussian quadrature formula. Double integrals using Trapezoidal and Simpson's rule.

Initial Value Problems For Ordinary Differential Equations: Single step Methods – Taylor Series, Euler and Modified Euler, Runge – Kutta method of order four for first and second order differential equations.Multistep Methods-Milne and Adam's Bashforth predictor and corrector methods.

Boundary Value Problems For Ordinary And Partial Differential Equations: Finite difference solution for the second order ordinary differential equations. Finite difference solution for one dimensional heat equation (both implicit and explicit), One-dimensional wave equation and two-dimensional Laplace and Poisson equations

Text Books:

1. Numerical Methods, Probability Theory and Statistics: A. K. Mukhopadhyay
2. Numerical Methods: Hilde and Brand

Object Oriented Analysis and Design	**MCA0024E**
4 - 0 - 0 : 4 Credits	Prerequisites: *Software Engineering*

Course Outcomes

At the end of the Course student will be

- Able to use an object-oriented method for analysis and design
- Able to analyse information systems in real-world settings and to conduct methods such as interviews and observations.
- Know techniques aimed to achieve the objective and expected results of a systems development process
- Know different types of prototyping, know how to use UML for notation

Syllabus

Object Oriented Design and Modelling: Object Oriented Fundamentals, Objects and object classes, object oriented design process, importance of modelling, principles of modelling, object oriented modelling.
Introduction to UML: Conceptual model of UML, building blocks of UML, Mechanisms in UML, architecture, software development life cycle.
Basic Structural Modelling Classes, relationships, common mechanisms, class and object diagrams. Advanced structural Modelling: Advanced classes, advanced relationships, Interfaces types and roles, packages, instances and object diagrams.
Collaboration Diagrams and Sequence Diagrams: Terms, concepts and depicting a message in collaboration diagrams. Terms and concepts in sequence diagrams. Difference between collaboration and sequence. diagram. Depicting synchronous messages with/without priority call back mechanism. Basic behavioural modelling: Interactions, use cases, Use Case Diagrams, Interaction Diagrams and activity diagrams.
Advanced behavioural modelling: Events and signals, state machines, process and threads, time and space, state chart diagrams. Architectural Modelling: Terms, Concepts, examples, Modelling techniques for component diagrams and deployment diagrams.

References:

1. Grady Booch, James Rum bough, Ivar Jacobson. 'The Unified Modeling Language User Guide. Pearson Edutaion 2002.
2. Ian Summerville, 'Software Engineering Sixth Edition' 2003.
3. Meilir Page Jones, 'Fundamentals of Object Oriented Design in UML', Addison Wesley.
4. The Elements of UML(TM) 2.0 Style, Scott W. Ambler, Cambridge University Press (May 9, 2005)
5. UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design , Jim Arlow&Ila
6. Neustadt, Addison-Wesley Professional; 2 edition (June 27, 2005)

7. Real Time UML Workshop for Embedded Systems, Bruce Powel Douglass, Newnes; Pap/Cdr edition (September 20, 2006)

Pattern Recognition	**MCA0025E**
4 - 0 - 0 : 4 Credits	Prerequisites: *None*

Course Outcomes

At the end of the Course student will be able to

- Explain and compare a variety of pattern classification, structural pattern recognition, and pattern classifier combination techniques.
- Summarize, analyze, and relate research in the pattern recognition area verbally and in writing.
- Apply performance evaluation methods for pattern recognition, and critique comparisons of techniques made in the research literature.
- Apply pattern recognition techniques to real-world problems such as document analysis and recognition.

Syllabus

Introduction: Pattern recognition and learning (supervised, unsupervised), training and test sets, feature selection; **Supervised learning and classification:** Discriminant functions and decision boundaries Linear discriminant functions, relaxation procedure, non-separable behavior Minimum distance classifier. Bayesian decision theory. Maximum likelihood classification. Parameter estimation, sufficient statistics, component analysis and discriminants (PCA, Fisher's) Nonparametric techniques. Density estimation, Parzen window, K-NN estimation, Supervised neural learning: Back-propagation algorithm, Radial basis-function neural net; Support vector machine classifier, Learning vector quantization

Unsupervised learning and clustering: Data description and clustering –similarity measures, criterion for clustering, Methods of clustering - partitional, hierarchical, graph theoretic, density based, k-means, k-mediod, fuzzy c-means clustering, Cluster validity

Feature extraction and feature selection: Problems of dimensionality- Feature extraction --PCA-Feature selection –KarhunenLoeve, stochastic approximation, kernel approximation, divergence measures, Independent component analysis

References:

1. R. O. Duda, P. E. Hart and D. G. Stork, Pattern Classification and Scene Analysis, 2nd ed., Wiley, New York, 2000.
2. J. T. Tou and R. C. Gonzalez, Pattern Recognition Principles, Addison-Wesley, London, 1974.
3. A. Konar, Computational Intelligence: Principles, Techniques, and Applications, Springer 2005

Social Networks	**MCA0026E**
4 - 0 - 0 : 4 Credits	Prerequisites: *None*

Course Outcomes

At the end of the Course student will be introduced to

- Understand a broad range of network concepts and theories.

- Appreciate how network analysis can contribute to increasing knowledge about diverse aspects of society.
- Use a relational approach to answer questions of interest to them. Analyse social network data using various software packages.
- Present results from social network analysis, both orally and in writing.

Syllabus

Introduction
Handling Real-world Network Datasets
Strength of Weak Ties& Homophily, +Ve / -Ve Relationships
Link Analysis
Cascading Behaviour in Networks
Power Laws and Rich-Get-Richer Phenomena and Epidemics
Small World Phenomenon
Pseudocore (How to go viral on web)

SUGGESTED READING:

1. Networks, Crowds and Markets by David Easley and Jon Kleinberg, Cambridge University Press, 2010
2. Social and Economic Networks by Matthew O. Jackson, Princeton University Press, 2010.

Soft Computing	**MCA0027E**
4 - 0 - 0 : 4 Credits	Prerequisites: *None*

Course Outcomes

At the end of the Course student will be introduced to

- Recognise the feasibility of applying a soft computing methodology for a particular problem.
- Apply fuzzy logic and reasoning to handle uncertainty and solve engineering problems.
- Apply genetic algorithm to combinational optimization problems.
- Apply neural networks to pattern classifications and regression problems. And compare solutions by various soft computing approaches for a given problem.

Syllabus

Introduction to Soft Computing, Introduction to Fuzzy logic,Fuzzy membership functions,Operations on Fuzzy sets; Fuzzy relations, Fuzzy propositions, Fuzzy implications, Fuzzy inferences

Defuzzyfication Techniques, Fuzzy logic controller; Solving optimization problems, Concept of GA, GA Operators: Encoding, GA Operators: Selection, Crossover, Mutation;

Introduction to EC, MOEA Approaches: Non-Pareto, Pareto;

Introduction to ANN, ANN Architecture; ANN Training, Applications of ANN;

REFERENCES:

1. An Introduction to Genetic Algorithm Melanic Mitchell (MIT Press)
2. Evolutionary Algorithm for Solving Multi-objective, Optimization Problems 2nd Edition), Collelo, Lament, Veldhnizer (Springer)
3. Fuzzy Logic with Engineering Applications Timothy J. Ross (Wiley)

4. Neural Networks and Learning Machines Simon Haykin (PHI)

Software Project Management	**MCA0028E**
4 - 0 - 0 : 4 Credits	Prerequisites: *None*

Course Outcomes

At the end of the Course student will be introduced to

- Identify the different project contexts and suggest an appropriate management strategy.
- Practice the role of professional ethics in successful software development.
- Identify and describe the key phases of project management.
- Determine an appropriate project management approach through an evaluation of the business context and scope of the project.

Syllabus

Software components - COTS and infrastructure - Software variability management- Software architecture design methods - Architecture evaluation and assessment methods - architectural styles. Design Patterns - Evolution patterns - Software artifact evolution processes - Case studies - Java Beans.

Product, Process and Project – Definition – Product Life Cycle – Project Life Cycle Models

Format Process Models And Their Use -Definition and Format model for a process – The ISO 9001 and CMM Models and their relevance to Project Management –Emerging Models - People CMM-Metrics – Configuration Management – Software Quality Assurance – Risk Analysis

Engineering and People Issues in Project Management-Phases (Requirements, Design, Development, Testing, Maintenance, Deployment) –Engineering Activities and Management Issues in Each Phase – Special Considerations in Project Management for India and Geographical Distribution Issues

RERERENCES:

1. Len Bass, Paul Clements, and Rick Kazman, "Software Architecture in Practice", 2nd Edition, Addison-Wesley Longman, Inc., Reading, MA, 2003
2. Richard N.Taylor, NenadMedvidovic, and Eric M.Dashofy, “Software Architecture: Foundations, Theory and Practice”, Wiley India Edition, 2012
3. Mary Shaw, and David Garlan ,” Software Architecture in Practice: Perspectives on an Emerging Discipline", PHI Learning Private Limited,2010
4. Ramesh and Gopalaswamy, "Managing Global Projects", Tata McGraw Hill,2001.

Speech and Natural Language Processing	**MCA0029E**
4 - 0 - 0 : 4 Credits	Prerequisites: *None*

Course Outcomes

At the end of the Course student will be introduced to

- How key concepts from NLP are used to describe and analyze language

- POS tagging and context free grammar for English language
- Understanding semantics and pragmatics of English language for processing
- Writing programs in Python to carry out natural language processing.

Syllabus

Introduction and Basic Text Processing
Spelling Correction, Language Modeling
Advanced smoothing for language modeling, POS tagging
Models for Sequential tagging – MaxEnt, CRF
Syntax – Constituency Parsing
Dependency Parsing
Lexical Semantics
Distributional Semantics
Topic Models
Entity Linking, Information Extraction
Text Summarization, Text Classification
Sentiment Analysis and Opinion Mining

TCP/IP Network Programming	**MCA0030E**
4 - 0 - 0 : 4 Credits	Prerequisites: *None*

Course Outcomes

At the end of the Course student will be introduced to

- TCP/IP Network Programming, Describe the TCP/IP protocol suite, Describe Internet addressing.
- Describe the services at the IP (Internet Protocol) layer, Describe the auxiliary protocols that serves the IP layer.
- Explain error reporting and query mechanism in the Internet, Describe broadcasting mechanism in the Internet, Identify routing protocols in the Internet.
- Describe process-to-process communication, Describe the application protocols that use the services of other layers, Explain the rationale for changes in the next generation of Internet protocols.

Syllabus

Introduction to Socket Programming – Overview of TCP/IP Protocols –Introduction to Sockets – Socket address Structures – Byte ordering functions – address conversion functions – Elementary TCP Sockets – socket, connect, bind, listen, accept, read, write, close functions – Iterative Server – Concurrent Server.

TCP Echo Server – TCP Echo Client – Posix Signal handling – Server with multiple clients – boundary conditions: Server process Crashes, Server host Crashes, Server Crashes and reboots, Server Shutdown – I/O multiplexing – I/O Models – select function– shutdown function – TCP echo Server (with multiplexing) – poll function – TCP echo Client (with Multiplexing).

Socket options – getsocket and setsocket functions – generic socket options – IP socket options – ICMP socket options – TCP socket options – Elementary UDP sockets – UDP echo Server – UDP echo Client – Multiplexing TCP and UDP sockets – Domain name system – gethostbyname function – Ipv6 support in DNS – gethostbyadr function – getservbyname and getservbyport functions.

Ipv4 and Ipv6 interoperability – threaded servers – thread creation and termination–TCP echo server using threads – Mutexes – condition variables – raw sockets – raw socket creation – raw socket output – raw socket input – ping program – trace route program.

References

1. W. Richard Stevens, TCP/IP Illustrated, Volume 1: The Protocols
2. W. Richard Stevens, Unix Network Programming, Volume 1: The Sockets Networking API
3. *Java Network Programming*, 2'nd ed., by Hugues, Shoffner, and Hamner, read chapters 1-21 and Appendix A.
4. Introduction to Web Server Development
5. Java Programming examples Quick Reference
6. Charlotte, a web server for Windows © 2000, Stuart Patterson, ACM Crossroads, May 2000.

Web Technology	**MCA0031E**
4 - 0 - 0 : 4 Credits	Prerequisites: *None*

Course Outcomes

At the end of the Course student will be introduced to

- Understand the principles of creating an effective web page, including an in-depth consideration of information architecture.
- Become familiar with graphic design principles that relate to web design and learn how to implement theories into practice.
- Develop skills in analyzing the usability of a web site and develop basic programming skills using Javascript & DHTML.
- Understand how to plan and conduct user research related to web usability.

Syllabus

Introduction to Web, Web development strategies, Web applications, Working of Internet, Connections, TCP/UDP/IP, IP addressing, Ipv4 to Ipv6. ARP, RARP, DCHP, ICMP, HTTP, SMTP, and E-mail SNMP, Domain Name System (DNS), Internet Services, WWW, Web Servers.
Web Browsers.
HTML and DHTML HTML Tag, Rules of HTML, Text Formatting & Style, List, Adding Graphics to Html Document, Tables and Layout , Linking Documents, Frame, Forms, Project in HTML.
Introduction to DHTML, CSS, Class & DIV, External Style Sheet.
XML: DTD, XML schemes, presenting and using XML.
Scripting: Java script: Introduction, documents, forms, statements, functions, objects; event and event handling; introduction to AJAX.
Introduction to active server pages (ASP), ASP.NET, java server pages (JSP), JSP application design, JSP pages, Session, Application: data base action.
PHP (Hypertext Preprocessor): Introduction, syntax, variables, strings, operators, if-else, loop, switch, array, function, form ,mail, file upload, session, error, exception, filter, PHP-ODBC.

References:

1. Xavier, C, " Web Technology and Design" , New Age International
2. Ivan Bayross," HTML, DHTML, Java Script, Perl & CGI", BPB Publication.
3. Ramesh Bangia, "Internet and Web Design" , New Age International
4. Bhave, "Programming with Java", Pearson Education
5. Ullman, "PHP for the Web: Visual QuickStart Guide", Pearson Education
6. Deitel, "Java for programmers", Pearson Education.

7. The complete Reference By Thomos A. Powell ,TMH publication
8. XML By Example, Sean McGrathPentice Hall Publication
9. Java Script :The definite Guide By Flangam .O'

Information System Security	**MCA0032E**
4 - 0 - 0 : 4 Credits	Prerequisites: *Computer Networks*

Course Outcomes

At the end of the Course student will be introduced to

- Exhibit knowledge to securecorrupted systems, protect personal data, and secure computer networksin an Organization.
- Practice with an expertise in academics to design and implement security solutions.
- Understand key terms and concepts in Cryptography, Governance and Compliance. Develop cyber security strategies and policies.
- Understand principles of web security and to guarantee a secure network by monitoring and analyzing the nature of attacks.

Syllabus

Basic Cryptography: - Classification of attacks, Evolution of Cipher Techniques, Symmetric and asymmetric key cryptography, Confusion and Diffusion.

Conventional Cryptography: - Substitution and Transposition ciphers. Cipher Implementation-P-Box, and Product Cipher, Fistal structure, Block Ciphers-DES, AES, Meet in the Middle Attack, Triple DES and IDEA, Classical Techniques, Modern Techniques, Algorithms, Confidentiality Using Conventional Encryption.
Public-Key Encryption and Hash Functions:- Public-Key Cryptography ,Deffie -Hellman Cryptosystem , Man in the Middle Attack ,Message Authentication and Hash functions; Hash and MAC Algorithms,SHA-1 Algorithm, RSA and the Knapsack algorithm.

Authentication Protocols:-Authentication techniques based on Shared Secret Key, Key Distribution Centre, Kerberos, Public Key Encryption and Public Key certificates. Digital Signatures:-Secret Key Signatures. Public Key Signatures and DSS.
Network Security Practice: Authentication applications, Kerberos, X.509 Directory Authentication Service, Electronic Mail Security; S/MIME,IP Security Architecture, Combining Security Associations, Key Management, Web Security; Web Security Requirements, Secure Sockets Layer and Transport Layer Security, Secure Electronic Transaction(SET), System Security: Intruders, Viruses and Related Threats, Types of Viruses, Trusted Systems.
E-mail and Internet Security: PGP and PEM, Firewalls, Types of Firewalls, Firewall Configuration, Firewall Design Principles, Classical attacks on the Internet, IP Sec, IP Spoofing attacks.

References:

1. William Stallings, Cryptography and Network Security, LPE Press.
2. A. Tanenbaum: Computer Networks, 3rd ed. Prentice Hall, 1996 (PHI 1997).
3. B. Schneider: Applied Cryptography, 2nd ed. Wiley, 1996.
4. C. Kaufman, R. Pearlman and M. Speciner: Network Security, Prentice Hall, 1995.
5. D. R. Stinson: Cryptography: Theory and Practice, CRC Press, 1995.
6. G. J. Simmons Ed.: Contemporary Cryptography, IEEE Press, 1991.
7. Behrouz A. Forouzen, Data Communication and Networking, TMH Press.

Advanced SoC Design	**MCA0033E**
4 - 0 - 0 : 4 Credits	Prerequisites: *None*

Course Outcomes

At the end of the Course student will be introduced to

- Arm Cortex-A processor architectures and Arm Cortex-A based SoCs, Capture the design of Arm Cortex-A based SoCs in a standard hardware description language
- Low-level software design for Arm Cortex-A based SoCs and high-level application development.
- Ability to use and choose between different techniques for digital system design and capture
- Ability to evaluate implementation results (e.g. speed, area, power) and correlate them with the corresponding high level design and capture.

Syllabus

SoC Design Concept Developed, Moore's Law, The Design Productivity Gap, Bridging the Design Productivity Gap, Example Arm-based SoC, Advantages of SoCs, Limitations of SoCs, SoC v Microcontroller v Processor, SoC Design Flow, SoC Example: NVIDIA Tegra 2, SoC Example: Apple SoC Families, Arm Processors and Applications, Arm Processor Families, Arm Processors vs Arm Architectures, Arm and Thumb Instruction Sets, AAPCS, Processormodes,Vector table ,Memory model, Memory types example CachedArmMacrobell, Data Alignment,Endianness, Coprocessors,PMU, Trust Zone, Virtualization, Arm Cortex-A Series Processors, Arm Cortex-A9 Processor, Cortex-A9 MP Core, What is NEON, NEON Registers.

Arm DS-5 Development Studio Overview, ARM DS-5 Code, ARM DS-5 Build, ARM DS-5 Debug, Debug Hardware, Virtual Debug Interface – VSTREAM, ARM DS-5 Analyser – Streamline, ARM DS-5 Analyser – Energy Probe, ARM DS-5 Simulation, ARM DS-5 Device Configuration Database, ARM assembler file syntax, Single/ Double register data transfer, Addressing Memory, Pre- and Post -Indexed Addressing, Multiple Register Data Transfer, Data Processing Instructions, Shift/Rotate Operations, Instructions for loading constants, Multiply/Divide, Bit Manipulation Instructions, Byte Reversal, Flow control, Branch instructions, Interworking, Compare and Branch if zero, Conditional Instructions, If Then, Coprocessor instructions, PSR access, DSP instructions overview, Saturated Maths and CLZ, Saturation, SIMD.

Cortex-A9 MPCore, Cortex-A9 MPE Configuration, Cortex-A9 Media Processing Engine, Register Renaming, Virtual Flags Registers, Small Loop Mode, Program Flow Prediction, Performance Monitoring Unit (PMU), Cortex A9 supports ARMv7-A Architecture, caches, Data Cache, Memory Management Unit, ARM v7 Architecture Effects, AMBA AX14-Lite, AXI Low Power Interface, GPIO Overview, AX14-Lite GPIO, Computer Memory, Memory Accessing, Volatile vs Non Volatile Memory, Types of Memory, Static RAM, Dynamic RAM, Non -Volatile Memory, Memory Controller, The Roles of a Memory Controller, Bus Types, Bus Terminology, Bus Operation, Communication Architecture Standards, ARM AMBA System Bus, AMBA 3 AXI Interface, AXI Components and Topology, Transaction Channels, Basic Signals, Clock and Reset, Channel Timing, Relationship Between the Channels

Serial Communication, Serial Communication vs Parallel Communication, Types of Serial Communication, UART Overview, UART Protocol, Character- Encoding Scheme, ASCII Encoded Characters, AXI UART Implementation, UART Control, UART Register Block, First In First Out(FIFO), Stream Data Transmission, AX-14 Stream Protocol, Data Streams, Global Signals, Master Signals, Slave Signals, Clock and Reset, Handshake, Packet Boundaries,VGA AXI4-LITE Overview, VGA Timing, VGA Interface, Utilization of FIFO, HDMI Overview, HDMI Interface, HDMI Signals: TMDS Channels, TMDS

Timing, Data Display Channels, Consumer Electronics Control, Hot Plug Detect, Edge Detection, Image Scaling, Gray Scale, Intensity GradientMagnitude, Software Programming: Edge Detection Algorithm.

Practical Implementations: Coding - Processing Text in Assembly Language; Design an GPIO Peripheral; Design an AXI4-Stream VGA-out Peripheral; Integrating Display Peripherals to Zynq SoC– HDMI to VGA Convertor; Accelerate Image Processing Using NEON; Accelerate Image Processing Using FPGA

Text Books and References:

- ARM System-on-Chip Architecture by Steve B. Furber
- ARM Assembly Language: Fundamentals and Techniques by William Hohl
- Cortex-A Series Programmer's Guide for ARMv7-A by Arm http://infocenter.arm.com/help/topic/com.arm.doc.den0013d/index.html

Advanced Computer Architecture	**MCA0034E**
4 - 0 - 0 : 4 Credits	Prerequisites: *None*

Course Outcomes

At the end of the Course student will be introduced to

- Perform computer arithmetic operations.
- Use the concepts and design of all type of sequential and combinational circuits.
- Design and conduct experiments, as well as to analyze of the hardware of a computer system and its components such as control unit, arithmetic and logical (ALU) unit, input/output, and memory unit.
- Be able to design techniques such as pipelining and microprogramming in the design of the central processing unit of a computer system.

Syllabus

Computer architecture definition and analyze historical and future computer architecture trends, illustrate the organization and function of components needed for a simple processor design, principles of instruction set design and demonstrate the use of Armv8-A Instruction Set Architecture, pipelining concepts, implementation, hazards, impact on performance, and application of them in the Arm10 processor pipeline.

Different methods of handling branches including dynamic branch prediction, predictors, branch target buffer, and their application in the Cortex-A15 processor, Working of exception handling, limits of pipelining, Compare and contrast the benefits and drawbacks of super pipelined and superscalar processor, approaches to Instruction-Level Parallelism (ILP) and components needed in a generic superscalar processor, Implementation of ILP concepts in the Cortex-A75, Cortex-A77, and Cortex-A55 processors.

Different types of memory and their comparison, memory hierarchy, working principle of Memory Management Unit (MMU), Translation Lookaside Buffers (TLBs), and Direct Memory Access (DMA), operation of MMU and TLBs work in the Cortex-A9 processor, purpose of a cache, different types of cache designs, cache policies, multi-level caches and cache performance metrics, with the Cortex-A9 processor as a case study, Usage of multicore processors by considering communication aspects, cache-coherence protocol and memory consistency, implementation of these aspects in the big.LITTLE combination in a DynamIQ cluster and Cortex-A55,

Operational concept of multithreading and the various approaches to multithreading. Compare Simultaneous Multithreading (SMT) and multicore, explore SMT feature in Cortex-A65, Compare and

contrast the functions and benefits of processor specialization such as vector processors, Graphics Processing Units (GPUs) and Single Instruction, Multiple Data (SIMD) architectures, Demonstration of the application of these specialization in Arm NEON, Scalable Vector Extension (SVE), Mali GPU G76, and Mali G77.

Text Books and References:

- •Digital Design and Computer Architecture Arm Edition by Sarah L. Harris & David Money Harris
- •ARM System-on-Chip Architecture by Steve B. Furber
- •Computer Organization and Design, The Hardware/Software Interface Arm Edition by David A. Patterson and John L. Hennesy
- •Computer Organization and Architecture Themes and Variations by Alan Clements

Embedded Linux	**MCA0035E**
4 - 0 - 0 : 4 Credits	Prerequisites: *None*

Course Outcomes

At the end of the Course student will be introduced to

- Embedded Linux operating system architecture.
- Linux-based embedded system component stack, Linux kernel modules.
- System configuration and boot process.
- Communication between kernel space and user space, System debugging and profiling.

Syllabus

Introduction to Embedded System, Embedded system components, Basic software, Operating systems for embedded systems, Linux-based embedded system components, Linux-based embedded system components, Reference hardware model, Reference hardware model implementations, CPU memory map, The role of the boot loader, The role of the boot loader, Possible scenarios, Possible scenarios, Linux kernel, Device tree, System programs and Application, Typical layout of the root file system.

Linux architecture, Conceptual view of the kernel, Process scheduler, Process scheduler, Memory manager, Memory manager external interfaces, Memory manager architecture, Virtual file system, i-node, i-node interface, File interface, Virtual file system architecture, Inter-process communication, Inter-process communication architecture, Network, Device trees, Build systems: Buildroot vs Yocto – general aspects, Buildroot vs Yocto – configuration, Buildroot vs Yocto – purpose, The Yocto Project, The Yocto build system, The build system workflow – configuration files, user configuration, Metadata, Machine (BSP) configuration, Distribution policy, image generation, SDK generation.

Introduction Linux kernel modules, CPU – I/O interface, CPU – I/O interface with polling, CPU – I/O interface with interrupt, CPU – I/O interface, CPU – I/O interface latency, CPU – I/O interface, Direct memory access (DMA) architecture, Direct memory access (DMA) transfer modes, I/O taxonomy, Typical operations, Linux devices, The Virtual File System (VFS): abstraction, functions – include/linux/fs.h, The device file concept, Linux kernel modules, The CPU/Device interface, The module level: file operations, octl() implementation, open()/release() implementation, read() implementation, Passing data to/from the kernel, Memory mapped I/O, GPIO-based I/O, The interrupt handler, Interrupt handling, Top-half and bottom-half, Needed support, Work queue, The user level, The user level – the application

Introduction to Application Demo: Building a Ranging Sensor Kernel Module, The sysfs file system – controlling GPIOs, Adding entries to the sysfs file system, Using sysfs and virtual file system API, The HC-SR04 ultrasonic ranging sensor, The HC-SR04 ultrasonic ranging sensor, Building Linux support for the HC-SR04 sensor, Module data structure definition, Module initialization function, Module clean-up

function, Module open function, Module close function, Module write function, Module read function, Module show and store function, Module test applications.

Practical Implementations: Introduction to the Board and Workspace set-Up; Custom Embedded Linux Build using the Manual Approach; Introduction to Linux kernel modules under Yocto; Handling general purpose i/o using Linux kernel modules; Handling HC-SR04 ranging sensor using Linux kernel modules; Introduction to code development and debugging using Yocto; Introduction to Linux kernel and application profiling

Text book titles and reference material

- Embedded Linux Systems with the Yocto Project by Rudolf K. Sterif
- Exploring Raspberry Pi: Interfacing to the Real World with Embedded Linux by Derek Molloy

Graphics and Mobile Gaming	**MCA0036E**
4 - 0 - 0 : 4 Credits	Prerequisites: *None*

Course Outcomes

At the end of the Course student will be introduced to

- Understand Arm Mali GPU architecture, Core OpenGL ES rendering techniques, Game design methodology
- Ability to use different graphics and game design techniques to optimize performance and reduce power consumption on mobile devices
- Ability to create computer graphics on mobile devices using the Open GL ES Library
- Ability to create 3D games from scratch using commercial game design engines.

Syllabus

Graphics Processing Overview, The Fundamentals of 3D Objects, 3D Coordinate Systems, Coordinate Space Conversation, Rendering Pipeline, Open GL and Open GL ES, Graphics Processing Units, A GPU is not a CPU, Example of Mali GPU-T880 GPU, Use of Game Engine, Module Structure-Gaming, GPU Architectures-Hardware Evolution, GPU Hardware Generations 1,2,3,4&5, Geometry Processor, GPU Architecture, GPU Design Principles, GPU Memory Systems, GPU Rendering Approaches, The Mali Family-Mali Graphics, Mali 55,200,300,400,450,T880 Overview, Midgard Family, Mali Rendering Pipline, Utgard Dataflow, Mali Midgrad Architecture.

Fixed Graphics Pipeline, Programmable shaders / Pipeline, Program Functions, Rendering Pipeline, Vertex Shader, Varyings, Viewport Clipping, Rasterizing, Fragment Shaders, Textures, Test and Blending, 3D Graphics, A 3D Object, Cube – Vertex Array, Types of Matrices- The Model Matrix, The View Matrix, The Projection Matrix, Identity Matrix, Matrix Addition, Matrix Multiplication, Translation, Scaling, Rotation, Texture Mapping – General Idea, UV Mapping, OpenGL ES and Textures, Lighting, Normals, Diffuse Light, Ambient Light, Specular Light, Lighting Strength, Video Games: A Historical Look, Video Games: From Then to Now, Introduction to Game Engines, The Current State of Game Engines, The Flagship Software, Games Development: The Coding Languages, Cocos2d-X: Open Source Development.

Complexity vs Performance, Methods for Increasing Performance and Complexity, Bump Mapping, Normal Mapping, Vertex Buffer Objects (VBO), VBO Functions, Mipmapping, Compressed Textures, Generating ETC Compressed Textures, ASTC Format, Graphical Assets, Loading Assets, Android Application Package – APK, Android File Loading, Extracting to a Private Location, Extracting to a Temporary Location, Software Development Process, Game Development Process: Iterative Development, Grouping Game Objects, Game Objects – 2D Games and Groupings, Breakdown the Design – Identifying

Components, Breakdown the Design - Update/Render Game Objects, Game Objects – 3D games, The Game Loop, Nodes, Scenes, Project Considerations.

Graphics – History of Hardware, Colour Cells – Commodore and NES, NTSC Artefact Colouring – Apple 2 Computers, Sprites, 2D Images – Characteristics, Polygons in Games, 3D Modelling, UV Mapping – Overview, Diffuse Mapping, Static Lighting - Light Mapping, Height Mapping, Normal/Bump/Displacement Mapping, Alpha Mapping, Cubemaps, Importance of external resources, Importance of proper sound design, Advanced 3D Effects, Animations – Production, Particle Effects, Particle Effects in the Game Loop, Bloom, 3D Camera Control, Introduction to Physics, Physics Bodies, AABB - Axis-Aligned Bounding Boxes, OBB – Orientated Bounded Boxes, Performance Optimisation, Introduction to Virtual Reality, Use of Virtual Reality, VR Headset, VR Software, Types of VR Systems, Working with VR , The VR Rendering Pipeline, The Fundamentals of Stereoscopic Rendering, Lenses, Distortion Shaders, Multiview rendering and multi-sampling, Calibrating Headtracking, Clock Locking, Bandwidth, Interaction and Controls, Augmented Reality.

Practical Implementations: Mali VR SDK Lab, Virtual Reality, Cocos2d-X And VR, Implementing VR libraries, Using Gear VR in our game

Text book titles and reference material

- Game Design Theory by Keith Burgun
- OpenGL ES 2 for Android: A Quick-Start Guide (Pragmatic Programmers) by Kevin Brothaler

Introduction to Robotics Systems	**MCA0037E**
4 - 0 - 0 : 4 Credits	Prerequisites: *None*

Course Outcomes

At the end of the Course student will be introduced to

- Demonstrate knowledge of the relationship between mechanical structures of industrial robots and their operational workspace characteristics.
- Demonstrate an ability to apply spatial transformation to obtain forward kinematics equation of robot manipulators, ability to generate joint trajectory for motion planning
- Demonstrate an ability to solve inverse kinematics of simple robot manipulators.
- Demonstrate an ability to obtain the Jacobian matrix and use it to identify singularities, knowledge of robot controllers.

Syllabus

Define the meaning of robotics, properties of robotic systems; List the features and benefits of the Arm Cortex-M7 processor, basic elements of an ARM-based (System on Chip) SoC design, Cortex-M7 processor core registers and their functions, functions of the Cortex-M7 processor components which includes NVIC, WIC, MPU, Bus Interconnect and Debug System, processor memory map and memory region, Endianness and the concepts of Little-endian and Big-endian, the ability to write simple Arm assembly code using the Arm instruction set, Arm instruction sets to write simple Arm assembly code.

Different types of interrupts and their functions, handling of interrupts and exceptions, difference between interrupts and exceptions, function of the Nested Vectored Interrupt Controller (NVIC) in low latency interrupt processing, function of exception mask registers, the concept of interrupt response latency and its importance, power supply requirements for autonomous cars, the functions and properties of a linear

voltage regulator, functions and operating principles of the boost converter, DC/DC converter and linear voltage regulator.

Purpose and function of sensors, distinguish between Op-Amp based inverting and non-inverting amplifier configurations, identify using circuit diagrams inverting and non-inverting amplifier, application of optical rotary encoders in sensing velocity, the key characteristics and elements of Robot Operating Systems(ROS), importance of (ROS) in developing robotics projects, ROS computation graph concepts such as node, topic, message, master, service.

A feedback control system for an autonomous car, effect of friction and drag and how they affect control system design, the concept of a non-holonomic system in relation to autonomous cars' operation, implementation of a closed loop steering control system, the key elements of a PID controller and best practice such as the Ziegler-Nichols method for tuning PID controller, Describe the importance of SLAM operation in autonomous robot's navigation, a generated map for autonomous robot navigation, identify elements that can contribute to uncertainty in robotic systems, Bayes filter algorithm and its application to calculate the robot's posterior probabilities.

Text Books and References:

- Robotics, Mechatronics, and Artificial Intelligence by Newton C. Braga
- Advanced Mechatronics and MEMS Devices by Dan Zhang
- Intelligent Mechatronic Systems: Modeling, Control and Diagnosis by Rochdi Merzouki and Arun Kumar Samantaray
- Robot Modeling and Control by Mark W. Spong, Seth Hutchinson, and M. Vidyasagar

Embedded Systems Design	**MCA0038E**
4 - 0 - 0 : 4 Credits	Prerequisites: *None*

Course Outcomes

At the end of the Course student will be introduced to

- Building Blocks of Embedded System
- Educate in Various Embedded Development Strategies, Bus Communication in processors, Input/output interfacing.
- Impart knowledge in various processor scheduling algorithms.
- Basics of Real time operating system and example tutorials to discuss on one real time operating system tool.

Syllabus

Introduction to Embedded Systems, CPUs vs. MCUs vs. Embedded Systems, Examples of, Embedded Systems, Options for Building Embedded Systems, Features of Embedded Systems, Introduction to Internet of Things (IoT), Challenges of IoT, Building Embedded Systems , Building Embedded System using MCUs, Introduction to the mbedTM Platform, Introduction to mbed, mbed Software Development Kit (SDK), , Hardware Development Kit (HDK), mbed Development Tools, mbed Worldwide Developer Community, Freedom KL25Z, NXP LPC1768 Hardware Platform, Nordic nRF51822 Hardware Platform, mbed and Internet of Things.

ARM Architectures and Processors, ARM Processor Families, ARM Cortex-M Series, Cortex-M0+ Processor, ARM Processor vs. ARM Architectures, ARM Cortex-M0+ Processor, Cortex-M0+ Block Diagram, Cortex-M0+ Memory Map, Bit-band Operations, ARM Cortex-M0+ Processor Instruction Set, ARM and Thumb Instruction Set, Cortex-M0+ Interrupts, NVIC (Nested Vectored Interrupt Controller),

Port Module and External Interrupts, Cortex Microcontroller Software Interface Standard (CMSIS), Benefits of CMSIS, CMSIS Functions, mbed Software Development Kit (SDK), Features of mbed SDK, mbed SDK Library Structure, Program Code, C Language vs. Assembly Language, Program-Generation Flow, Program Image, Program Data, Data Types, Accessing Data using C and Assembly, Mixed Assembly and C Programming, Embedded Assembly.

Digital Input and Output, Voltages and Logic Values, GPIO Controller, Using Pointer to Access, GPIO, Define Data Structure for Peripherals, Digital IO Examples, Using LED, Using 7-Segment Display, Using Infrared Emitter/ Detector, Analog Input, Digital-to-Analog Converter, Analog Output, Analog-to-Digital Converter, ADC range, Resolution and Quantization, Sampling Frequency, Input/ Output Analog Signals using mbed, mbed Analog Input, mbed Analog Output.
Timer and Pulse-Width Modulation, Timer Overview, Components of a Standard Timer, Compare Mode, Capture Mode, Pulse-Width Modulation Mode, mbed Timer and PWM, mbed Timer, mbed time ticker, mbed PWM, Serial Communication Overview, UART Communication, Operating System Overview, Types and Services of Operating Systems, Real-Time Operating System, RTOS Overview, RTOS Task Scheduling, Keil RTX RTOS, RTOS on mbed platform, mbed RTOS API, Using mbed RTOS API for your Project, Thread, Mutex and Semaphore.

Practical Implementation; Introduction to the Keil Mdk-Arm Tool, C And Assembly Coding - Processing Text In Assembly Language, Square Root Approximation, Digital Input/ Output and GPIO, Interrupt and Low Power Features, Programming using mbed API, Analog Input and Output Timer and PWM, Serial Communication

Text Books and References:

- The Definitive Guide to the ARM Cortex-M0 by Joseph Yiu
- Embedded Systems Fundamentals on Arm Cortex-M based Microcontrollers: A Practical Approach by Alexander G. Dean https://www.arm.com/resources/education/textbooks/efficient-embedded-systems
- White Paper: Cortex-M for Beginners - An overview of the Arm Cortex-M processor family and comparison: https://community.arm.com/developer/ip-products/processors/b/processors-ip-blog/posts/white-paper-cortex-m-for-beginners-an-overview-of-the-arm-cortex-m-processor-family-and-comparison.

Object Oriented Programming	**MCA0039E**
4 - 0 - 0 : 4 Credits	Prerequisites: *None*

Course Outcomes

At the end of the Course student will be introduced to

- Define the features of C++ supporting object oriented programming. Analyze the relative merits of C++ as an object oriented programming language.
- Derive the major object-oriented concepts like encapsulation, constructor, operator & function overloading.
- Major object-oriented concepts to implement inheritance and polymorphism
- Apply advanced features of C++ specifically stream I/O, and templates.

Syllabus

Introduction to OOP – Overview of C++ - Classes and Object, Structures, Friend Functions , Friend Classes, Inline functions .
Constructors ad Destructors –Dynamic Initialization of Objects - Static Members – Passing objects to functions – Function returning objects-Arrays of Objects, Object as Function Arguments;
Arrays, Pointers, this pointer, References, Dynamic memory Allocation;
Functions Overloading, Default arguments, Overloading Constructors, Pointers to Functions; Operator Overloading, Type Conversion;
Inheritance: Types, Derived Class Constructors, Issues in Inheritance; Virtual base Class;
Run-time Polymorphism: Virtual functions,Pure virtual functions
Class templates and generic classes, Function templates and generic functions;
Exception Handling: Derived class Exception, Exception handling Functions
Streams- File I/O, Name spaces –Array based I/O – Error handling during file operations - Formatted I/O;
STL: Overview-Container Classes Lists-Maps- Algorithms Using Functions and Objects-String Class - Sequence Containers, Iterators-Specialized Iterators - Associative Containers. Storing User- Defined Objects - Function Objects

Text Books:

1. Stephen Prata,"C++ Primer Plus", 6th Edition,Addison-Wesley Professional, 2011
2. BjarneStroustrup, "Programming: Principles and Practice Using C++,1st Edition, Addison-Wesley Professional, 2008
3. Andrew Koenig and Barbara E. Moo, "Accelerated C++: Practical Programming by Example", 1st Edition, Addison-Wesley Professional, 2000
4. Bruce Eckel ,"Thinking in C++: Introduction to Standard C++: Volume One" 2nd Edition,PrenticeHall, 2000
5. Andrei Alexandrescu , "Modern C++ Design: Generic Programming and Design PatternsApplied" , 1st Edition, Addison-Wesley Professional, 2001

Programming in Python	**MCA0040E**
4 - 0 - 0 : 4 Credits	Prerequisites: *None*

Course Outcomes

At the end of the Course student will be introduced to

- To understand why Python is a useful scripting language for developers.
- To learn how to design and program Python applications.
- To learn how to use lists, tuples, and dictionaries in Python programs.
- To learn how to identify Python object types.

Syllabus

The Context of Software Development-Software-Learning Programming with Python
Values and Variables-Integer and String Values-Identifiers-User Input-String Formatting
Expressions and Arithmetic-Expressions-Arithmetic Examples
Conditional Statements-Boolean expressions-If/Else statement-Other Conditional Expressions
Iteration-Loops
Using Functions-Introduction to Using Functions-Functions and Modules
Writing Functions- Function Basics -Parameter Passing-Custom Functions vs Standard Functions-Refactoring, Global Variables-Making Functions Reusable-Functions as Data
Objects-Using Objects-String, File Objects
Lists-Using Lists-Building Lists-List Traversal
Tuples, Dictionaries, and Sets-Storing Aggregate Data-Enumerating the Elements of a Data Structure

Inheritance and Polymorphism
Advanced Python Applications and Packages

Text Books and References:

1. Fundamentals of Python Programming, Richard L. Halterman
2. Kenneth A. Lambert, The Fundamentals of Python: First Programs, 2011, Cengage Learning,
3. Maste ring python for data science, Samir Madhavan