Assam Women's University, Rowriah, Jorhat-04

Syllabus of MCA programme

(Draft)

Department of Information & Computer Science
School of Engineering & Technology

About MCA Programme:

Name of the programme: Master of Computer Application (MCA)

Duration: Two year (minimum-4 semesters, maximum-6 semesters)

Eligibility criteria: Passed BCA/Bachelor Degree in Computer Science Engineering or equivalent Degree. OR Passed B.Sc/B.Com/B.A with Mathematics at 10+2 Level or at Graduation Level. Obtained at least 50% marks (45 %marks in case of candidates belonging to reserved category) in the qualifying examination.

Programme Objective: Objectives of the programme are as follows

PO1.To produce computer professionals with good ethical values and knowledge of computer applications, who can successfully design feasible and technically sound innovative solutions to real life problems.

PO2. equip students with all the skills required to develop application software and information system in diverse area where computers are used.

PO3. expose the students with latest software tools and new technologies.

PO4. The programme delves to inculcate an inquisitive mind to promote research-based thinking and encourages students to write research articles to step into the domain of Research and Development. It prepare Graduates who will contribute to societal growth through research in their chosen field.

PO5. prepare graduates who will perform both as an individual and in a team through good analytical, designing, implementation skills and with professional ethics.

Programme Outcomes: On completion of MCA degree, the graduates will be able to-

PO1. Apply the knowledge of computing and mathematics relevant to Information Technology to various real-life applications for any given requirement.

PO2. Identify, analyze, formulate and solve complex Computing problems reaching substantiated conclusions using fundamental principles of Mathematics, Computing sciences, and relevant domain disciplines.

PO3. Design and develop application software for any desired needs with appropriate considerations for any specific requirement on societal and environmental aspects.

PO4. Understanding and application of modern tool and technique: Create, select, adapt and apply appropriate techniques, resources and modern computing tools to computing activities with an understanding of the limitations.

PO5. Create systems through software development to solve problems in Industry domain areas.

PO6. Understand and commit to professional ethics, cyber regulations, responsibilities and norms of professional computing practice.

PO7. Involve in perennial learning for a continued career development and progress as a computer professional.

PO8. Create a culture that focus on innovation and Entrepreneurship.

PO9. Function effectively as an individual and as a member or leader in diverse teams and multidisciplinary environments, communicate effectively and present technical information in oral and written reports.

Courses Offered:

The courses offered in this programme is classified into the following:

- **Core Course**: There are total of 14 core courses of 4 credits or 2credits that are compulsory for the students. Students are required to study 4 core courses in semester I, II and 3 core courses in semester III and IV. In the IV semester, students shall submit a project of of 4 credit. A core course on "Gender Sensitization" is compulsory for the students to attend in I semester.

- **Discipline Specific Elective Course (DSE)**: The Department offers 16 DSE courses with 4 credit each. Students are required to choose 4 DSE from these; one each in I, II, III and IV semester.

- **Generic Elective (GE)**: Generic Elective courses are offered by other Departments. Generic Elective course is of 4 credits. Students are required to opt for 1 course each in III and IV semester.

- **Ability Enhancement Course**: There are 2 AEC of 2 credit each which are offered to students in I and II semesters. These are compulsory non-credit courses for students.

- **Skill Enhancement Course**: There are 2 SEC of 2 credit each which are offered to students in I and II semesters. These are compulsory non-credit courses for students.
- **Bridge Course:** Bridge courses are offered for those students who come from discipline other than Computer Science. Theses courses are of 2 credits and non-cgpa mandatory course for non computer science background students.

Distribution of courses

(name, code and credit(L+T+P))

Semester	CGPA mandatory Course			Non CGPA Mandatory Course			Total credit
	Core courses (credit=4 or 2)	***Discipline Specific Elective Courses (DSE) (Credit =4)***	***Generic Elective Courses(GE) (credit =4)***	***Skill Enhancement Course (credit=2)***	***Ability Enhancement Course (credit=2)***	***Bridge Course*** (Mandatory for only those students with graduation other than Compute Science Discipline)	
I	Discrete Mathematics (MCAC1401) (3+1+0) Computer Organization and Architecture (MCAC1402) (2+1+1) Introductory Computing using C (MCAC1403) (2+1+1) Gender Sensitization (1+1+0)	DSE-1		Basic Web Technology (MCAS1201) (1+0+1)	Linux Administration (MCAA1201) (1+0+1)	Fundamentals of Computer (MCAB1301) (1+1+1)	18
II	Data Communication and Computer Networks (MCAC2401) (2+1+1) Data Structures using c++ (MCAC2402) (2+1+1) Database Management System (MCAC2403) (2+1+1)	DSE-2	-	Introduction to Latex and R Programming (MCAS2201) (0+0+2)	Introduction to Python programming (MCAA2201) (1+0+1)		20

	Operating System (MCAC2404) (2+1+1)						
III	Advanced Web Technology (MCAC3401) (2+1+1)	DSE-3	GE-1	-			0
	Object Oriented Programming and Design using Java (MCAC3402) (2+1+1)						
	Introduction to Artificial Intelligence (MCAC3403) (3+1+0)						
IV	Data Mining (MCAC4401) (2+1+1)	DSE-4	GE-2	-			20
	Software Engineering (MCAC4402) (2+1+1)						
	Project (MCAC4403) (0+0+4)						
Total credit							**78**

Credit Distribution of Courses

Semester	Credit of CGPA Mandatory Courses			Credit of Non CGPA Mandatory Courses			Total Credit
	Core courses	Discipline Specific Elective Courses	Generic Elective Courses	Ability Enhancement Courses	Skill Enhancement Courses	Bridge Course	
I	3x4	1x4	-	1x2	1x2	1x3	18
	1x2	-	-	-	-		
II	4x4	1x4	-	1x2	1x2		20
III	3x4	1x4	1x4	-	-		20
IV	3x4	1x4	1x4	-	-		20

Total Credits: 78

List of Discipline Specific Elective Courses (DSE)*

Semester	Sl. No.	Course code	L+T+P	Course title
I	1	MCAE1401	3+1+0	Graph Theory
	2	MCAE1402	2+1+1	Digital Logic and Design
	3	MCAE1403	2+1+1	Numerical Methods
	4	MCAE1404	3+1+0	Formal Language and Automata
II	5	MCAE2401	3+1+0	Design and Analysis of Algorithms
	6	MCAE2402	2+1+1	Computer Graphics
	7	MCAE2403	3+1+0	Compiler Design
	8	MCAE2404	2+1+1	Embedded Systems
III	9	MCAE3401	3+1+0	Computer Security and Cryptography
	10	MCAE3402	2+1+1	Introduction to Machine Learning
	11	MCAE3403	3+1+0	Advanced Operating System
	12	MCAE3404	3+1+0	Image Processing
IV	13	MCAE4401	2+1+1	Data Analytics with Python
	14	MCAE4402	3+1+0	Natural Language Processing
	15	MCAE4403	3+1+0	Block Chain Architecture and its use cases
	16	MCAE4404	2+1+1	Android Application Development

*Student has to choose one discipline specific elective course out of four discipline specific elective courses in each semester.

Detailed Syllabus of each course is given below. Each syllabus starts with the title of the course and its code. Course contents are divided into modules. Contact hours of each module is shown against it in brackets.. A brief description on practical component of a course is also given along with contact hours in parenthesis.

Core Courses

Discrete Mathematics
Computer Organization and Architecture
Introductory Computing using C
Data Communication and Computer Networks
Data Structures
Database Management System
Operating System
Advanced Web Technology
Object Oriented Programming and Design
Artificial Intelligence
Data Mining

Software Engineering Project

Discrete Mathematics
MCAC1401

1. About the Course

This is a Core Course about the mathematical language of computer science. Concepts and notations from discrete mathematics are useful in studying and describing objects and problems in all branches of computer science, such as design and analysis of algorithms, programming languages, automata theory, Artificial Intelligence, Data Science, software development etc. Conversely, computer implementations are tremendously significant in applying ideas from discrete mathematics to real-world applications, such as operations research.

2. Course Description

- Target Audience:
  - 1st semester students of MCA programme
- Course Period: One semester
- Total Credit (L + T + P): 4 (3 + 1 + 0)
- Total Credit hours: 4
- Total Contact Hours: 56 Hours (14 Weeks X 4 Hours)
  - Lectures: 42 Hours (14 Weeks X 3 Hours)
  - Tutorial: 14 Hours (14 Weeks x 1 Hour)
  - Practical: Nil

2.1 *Prerequisites and Dependencies*

There is no particular prerequisite for the course. However, the students are expected to have elementary knowledge of general mathematics and logical reasoning.

2.2 *Objective*

The objective of this course is to explain the basic theory of discrete mathematics applied in different fields of Computer Science. Concepts earned in this course will be used in subsequent courses such as “Design and Analysis of Algorithms”, “Formal Language and Automata”, “Software Engineering”, “Introduction to Machine Learning” , “Introduction to Artificial Intelligence”, “Data Mining” etc.

2.3 *Course Outcomes*

After course completion, following are the course outcomes.

CO1. Students completing this course will get understanding of the concepts of set, function, relation, different algebraic structures, determinant and matrices.

CO2. Students completing this course will be able to apply the rules of inference and methods of proof including direct and indirect proof forms, proof by contradiction, and mathematical induction.

CO3.Students completing this course will be able to use tree and graph algorithms to solve problems.

CO4. Students completing this course will be able to evaluate Boolean functions and simplify expressions using the properties of Boolean algebra.

CO5. Students completing this course will be able to apply the knowledge of discrete probability and statistics in different domain of computer science.

CO6. Students will be able to solve system of linear equation.

CO6. Students will be able to solve problems involving recurrence relations and generating functions.

3. Course Contents

Module I: Basic Structures **(8 hours)**

Sets, Set Operations, Functions, Sequence and Summation, Cardinality of sets, Relations and their properties, Representing relations, Closure of relations, Equivalence relations, Partial ordering, Algebraic structure-group, ring, field

Module II: Logic and Proofs **(9 hours)**

Propositional logic and its application, Propositional equivalence, Predicates and quantifiers, Nested quantifiers, Rules of inference, Boolean algebra, Introduction to proofs, Proof methods and strategy, Mathematical induction, Recursive definition and structural induction.

Module III: Number Theory **(7 hours)**

Divisibility and modular arithmetic, Primes and Greatest Common Divisors, Solving congruence, Application of congruence.

UNIT IV: Counting **(7 hours)**

Basics of counting, The Pigeonhole principle, Permutation and combination, Application of recurrence relation, solving linear recurrence relation, Generating functions.

UNIT V: Probabilitiy and statistical concept **(9 hours)**

An introduction to discrete probability, Probability theory, Conditional Probability, Baye's theorem, Mean, median, mode, Random variable and distribution, Expected values and variance.

UNIT VI: Graphs and Trees **(7 hours)**

Graphs and graphs models, Graph terminology, Special types of graphs Graph isomorphism, Walk, Path, Cycle, Connectivity, Hamiltonian path, shortest path algorithm, Introduction to trees, Tree traversal, Spanning tree, Minimum spanning tree.

Unit VII: Matrices and solution of linear equations **(9 hours)**

Determinant and matrices, matrix inversion, rank of matrices, system of linear equation, solution of linear equation

4. Referential Sources

Book:

1. Rosen K.H., "Discrete Mathematics and it's applications". McGraw Hill
2. Seymour Lipschutz, Marc Lars Lipson, "Discrete Mathematics". SCHAUM'S ouT lines
3. Narsingh Deo, "Graph Theory with Application to Engineering and Computer Science". PHI Publishing.
4. Jay L.Devore," Probability and Statistics for Engineering and the Sciences".Cengage India Private Limited

Computer Organization and Architecture
MCAC1402

1. About the Course

This is a Core Course which deals with the structure of different parts of a computer system and how these parts function together to form the whole system. Low level programming of a system for a specific architecture is also addressed in this course.

2. Course Description

- Target Audience:
  - 1st semester students of MCA programme only.
- Course Period: One semester (14 Weeks)
- Total Credit (L + T + P): 4 (2 + 1 + 1)
- Total Credit Hours: 5 hours
- Total Contact Hours: 70 Hours (14 Weeks X 3 hours + 14 weeks x 2 hours)
  - Lectures: 28 Hours (14 Weeks X 2 Hours)
  - Tutorial: 14 hours (14 Weeks x1 Hour)
  - Practical: 28 hours (14 weeks x 2 hours)

2.1 *Prerequisites and Dependencies*

There is no prerequisite for this course. Students are expected to have fundamental knowledge on computer hardware s, software and programming.

2.2 *Objective*

The objective of the course is to equip students with the knowledge of structure and function of different units of a computer system and how these units coordinate with each other to perform a task. The students are also taught about Instruction set Architecture and assembly language programming.

2.3 *Course Outcomes*

After course completion, following are the learning/course outcomes.

CO1. To understand the internal structure of a computer system.

CO2. To describe structure of different units of a computer system and how they are organized and co-ordinate with each other.

CO3. To understand Instruction Set Architecture and machine language.

CO4. To write assembly language program.

3. Course Contents

Module I: Introduction to Digital system (5 hours)

Digital vs Analogue System, Binary number system, 1's and 2's complement, Information representation using code, ASCII code, Digital logic gates, Combinational circuit-adder, subtractor, Decoder, Encoder, MUX, demultiplexer, Sequential Circuit-flipflop, registers

Module I: Basic structure of computer (3 hours)

Block diagram of a Computer System, Functional Units of computer, Basic Operational Concept, Bus Structure, Performance-processor clock, performance equation, pipelining and Superscalar Operation, Instruction set: CISC and RISC, Multiprocessor and Multicomputer, Computer generations

Module III: Machine Instructions and Programs (9 hours)

Number, arithmetic operations and characters, Memory locations and addresses-Byte addressability, Big-Indian and Little-Indian assignment, Word Assignment, Memory operations, Instructions Register transfer notation, Assembly language notation, Basic instruction type, Instruction execution, Branching, Addressing modes, Assembly language-assembler directive, Basic Input-Output Operation, Stacks, Subroutines

Module II: The Memory System (9hours)

Some basic concepts, Interfacing memory with processor, Semiconductor RAM Memories, Internal Organization of Memory Chips, Static and Dynamic memory System, Memory Controller, Read Only Memories Speed, Size and Cost of Memory, Cache Memories-Mapping function and replacement algorithm, Hit Rate and Miss Penalty, Virtual memories-address translation

Module IV: Input / Output Organization (8 hours)

Accessing input/output device, interrupts-interrupt hardware, enabling and disabling interrupt, handling multiple devices, controlling device request, Exceptions, Direct Memory Accesses, Buses-Synchronous and asynchronous buses, Interface circuits-parallel and serial port, Standard I/O interface-PCI bus, SCSI bus, USB bus.

Module V: Pipelining (8 hours)

Pipeline performance, data hazards-operand forwarding, handling data hazards in software, Instruction hazards- unconditional branches, Conditional branches and branch prediction, Influence on instruction set, Design Issue.

4. Laboratory Sessions (28 hours)

The practical component of this course is of one credit, i.e., 2 hours of classes per week. The laboratory sessions mainly deals with the following topics-

1. Introduction to 8086- Instruction Set Architecture and the simulator to be used for programming.
2. Assembly language programming and assembler.
3. Different types of registers.
4. Memory addressing.
5. Variable, array, constant etc.
6. Data transfer operation.
7. Interrupt and I/O operations
8. Arithmetic and logic instructions
9. Program flow control- branching, looping etc

4. Referential Sources

Books:

- Carl Hamacher, Zvonko Vranesic, Zaky, “Computer Organization”, McGraw Hill
- William Stallings,” Computer Organization and architecture: Designing for performance”, Pearson Education India
- Mano M Morris, “ Computer System Architecture”, Pearson Education India

Introductory Computing using C
MCAC1402

1. About the Course

This is a *Core Course* and is aimed at advancing concepts of programming and software code organization within the framework of structural and procedural programming paradigms. The course is organized as a series of lectures and hands-on laboratory sessions using C programming languages and focusing on discussing how to write a program of moderate complexity by using C language.

2. Course Description

- Target Audience:
  - 1st semester students of MCA programme only.
- Course Period: One semester (14 Weeks)
- Total Credit (L + T + P): 4 (2 + 1 + 1)
- Total Credit Hours: 5 hours
- Total Contact Hours: 70 Hours (14 Weeks X 3 hours + 14 weeks x 2 hours)
  - Lectures: 28 Hours (14 Weeks X 2 Hours)
  - Tutorial: 14 hours (14 Weeks x1 Hour)
  - Practical: 28 hours (14 weeks x 2 hours)

2.1 *Prerequisites and Dependencies*

There is no prerequisite to this course.

2.2 *Objective*

The course is oriented to those who want to advance structured and procedural programming understating and to improve C programming skills. The major objective is to provide students with understanding of code organization and functional hierarchical decomposition with using complex data types.

2.3 *Course Outcomes*

After course completion, following are the learning/course outcomes.

CO1. Understanding a functional hierarchical code organization.
CO2. Ability to work with textual information, characters and strings.
CO3. Ability to work with arrays of complex objects
CO4. Understanding a concept of object thinking within the framework of functional model.
CO5. Ability to handle possible errors during program execution.

3. Course Contents

Module I: Introduction to computer and programming **(6 hours)**

Computer, block diagram of computer, hardware, software, program, types of software, operating system, compiler, programming languages, types of programming languages, algorithm, pseudocode, flowchart, desirable characteristics of a program, C programming language, history of C, structure of a C program, C character set, identifiers and keywords, writing, compiling and executing a c program, datatypes, constants, escape sequences, string constants, variables and arrays, declarations, expressions, statements, symbolic constants

Module II: Operators and I/O **(5 hours)**

Arithmetic operators, unary operators, relational and logical operators, assignment operators, conditional operator, data input output, single character input, single character output, data input from user: scanf function, writing output data: printf function, display formatting using printf function, the gets and puts function, error and debugging techniques

Module III: Control statements **(6 hours)**

Control statements, branching statement, looping statements: for, while, do-while, switch statement, break statement, continue statement, goto statement

Module IV: Functions and arrays **(6 hours)**

C functions, defining a function, calling a function, function prototypes. Passing arguments to function, recursion, defining an array, processing an array, passing array to a function, multidimensional array

Module V: Storage classes, strings and pointers **(6 hours)**

Storage classes, automatic variables, register variables, external variables, static variables, defining and initializing a string, null character, reading and writing a string, library functions for strings, pointers, pointer declaration, passing pointer to function, pointers and one-dimensional array, dynamic memory allocation, operations on pointers, pointers and multidimensional array, arrays of pointers, passing functions to other functions

Module VI: Structures and unions **(5 hours)**

Structure, defining a structure, processing structure, user defined data types, structure and pointers, passing structures to functions, self-referential structures, union, defining a union, processing union

Module VII: Handling file **(4 hours)**

Files, opening and closing files, reading and writing a file, processing a file, unformatted files, binary files, random access of files

Module VIII: Bitwise operators and bitfield **(4 hours)**

Bitwise operators, one's complement operator, logical bitwise operators, masking, shift operators, bitwise assignment operators, bitfields, defining bitfields, processing bitfields

4. Laboratory Sessions **(28 hours)**

The practical component of this course is of one credit which amounts to 2 hours of Laboratory class per week.

- Introduction to the concept of Keywords, Datatypes, Operators, Variables, Constants in C.
- Branching statements
- Control statements using loops and goto command
- Introduction to Functions
- Introduction to Arrays
- Introduction to Pointers in C.
- Strings
- Structures and Unions
- File Handling

5. Referential Sources

Books:

- Byron Gottfried, "Programming with C". McGraw Hill Education
- S.K. Srivastava, "C in Depth". BPB Publications

Useful Web Sources:

- http://www.cprogramming.com/ : C Programming and C++ Programming

---

Data Communication and Computer Networks MCAC2401

1. About the Course

This is a *Core Course* and is aimed to make the students understand the concepts of computer network and make them capable of configuring networks and handling issues that arise in a computer network.

2. Course Description

- Target Audience:
  - 2nd semester students of MCA programme only.
- Course Period: One semester (14 Weeks)
- Total Credit (L + T + P): 4 (2 + 1 + 1)
- Total Credit Hours: 5 hours
- Total Contact Hours: 70 Hours (14 Weeks X 3 hours + 14 weeks x 2 hours)
  - Lectures: 28 Hours (14 Weeks X 2 Hours)
  - Tutorial: 14 hours (14 Weeks x1 Hour)
  - Practical: 28 hours (14 weeks x 2 hours)

2.1 *Prerequisites and Dependencies*

Computer Network has no dependency however, a student needs to have knowledge about C programming as the practical classes of the course covers an advanced networking module of C programming that is called *Socket programming*.

2.2 *Objective*

The major objective of this course is to provide students with understanding of all the aspects of a communication over a network and all the standards related to such communication. This course presents understanding of entire communication process and standards between a pair of programs under execution (processes). The course also includes lab component that covers the programming aspect to develop such programs.

2.3 *Course Outcomes*

After course completion, following are the learning/course outcomes.

CO1. Understanding TCP/IP and OSI model
CO2. Understanding signal, transmission, transmission media and switching
CO3. Understanding TCP/IP model and OSI model
CO4. Understanding the Data Link layer in TCP/IP model.
CO5. Understanding the Network layer in TCP/IP model.
CO6. Understanding the Transport layer in TCP/IP model.
CO7. Understanding application layer

3. Course Contents

Module I: Introduction **(4 hours)**

Data communications, components, data representation, data flow, networks, physical structure of network, network models, categories of networks, Internetwork, Internet, protocols, OSI model, layers in the OSI model: physical layer, data link layer, network layer, transport layer, session layer, presentation layer, application layer, TCP/IP protocol suite, comparison between OSI and TCP/IP model, physical addresses, logical addresses, port addresses

Module II: Signals and transmission **(8 hours)**

Analog and digital data, analog and digital signals, periodic and non-periodic signals, sine wave, phase, wavelength, time and frequency domains, composite signals, bandwidth, digital signal, bit rate, bit length, digital to digital conversion, line coding, block coding, analog to digital conversion, pulse code modulation, delta modulation, parallel and serial transmission, digital to analog conversion, amplitude shift keying, frequency shift keying, phase shift keying, quadrature amplitude modulation, analog to analog conversion, amplitude modulation, frequency modulation, phase modulation

Module III: Transmission media and switching **(5 hours)**

Guided media, twisted pair cable, co-axial cable, fiber-optic cable, unguided media, radio waves, micro waves, infrared, switching-packet, message and circuit switching, circuit

switched networks, datagram networks, virtual circuit network, telephone network, dial up modems, DSL

Module IV: Data link layer **(6 hours)**

Data link layer and its functionalities , types of errors, redundancy, detection versus correction, coding, polynomial code, block coding, hamming distance, linear block codes, cyclic codes, CRC, checksum, framing, flow control, error control, protocols, simplest protocol, stop and wait protocol, go back n protocol, selective repeat protocol

Module V: Network layer **(7 hours)**

Network layer and its functionalities, IPv4 addresses, address space, notation of IP addresses, classful addressing, classless addressing, IPv6 addresses, address mapping, ARP, RARP, DHCP, forwarding techniques, forwarding process, Adaptive and nonadaptive routing, unicast routing protocols, distance vector routing, link state routing, path vector routing, unicast, multicast and broadcast routing

Module VI: Transport layer **(6 hours)**

Transport layer and its functionalities, Process to process delivery, client/server paradigm, multiplexing and demultiplexing, connectionless versus connection-oriented service, TCP handshaking, connection establishment and connection release, data traffic, congestion, congestion control, QoS, integrated services, differentiated services

Module VII: Application layer **(6 hours)**

Domain name system, namespace, flat namespace, hierarchical name space, domain name space, Internet, DNS in Internet, name resolution, mapping between names and addresses, telnet, electronic mail, FTP, WWW, browser, web server, URL, cookies, Application layer protocol-SMTP, MIME, a brief introduction to cryptography, symmetric vs asymmetric cryptography

4. Laboratory Sessions (28 hours)

This course contains 2 hours of practical classes per week. Following topics will be covered in the laboratory classes.

- Introduction to switches, routers, cables.
- Crimping of cat 6 cables.
- Various networking commands in windows operating systems.
- Various networking commands in linux operating systems.
- Configuring a LAN using switches and cat 6 cable.
- Socket programming in C.

5. Referential Sources

Books:

- Forouzan Behrouz A., “Data Communications and Networking”. Mcgraw Hill
- Stallings William. “Data and Computer Communications”. Pearson Education India

Data Structures using c++

MCAC2402

1. About the Course

This is a *Core Course* and is aimed at teaching efficient storage mechanism of data for an organized and easy access using one of the most fundamental programming language that is C++. It is so essential that all companies dealing with system software as well as application software development have a need for the same. The course will focus on various idioms of C++ and attempt to explore every C++ feature justifying and illustrating them with several examples and assignments.

2. Course Description

- Target Audience:
  - 2nd semester students of MCA programme only.
- Course Period: One semester (14 Weeks)
- Total Credit (L + T + P): 4 (2 + 1 + 1)
- Total Credit Hours: 5 hours
- Total Contact Hours: 70 hours(14 Weeks X 3 hours + 14 weeks x 2 hours)
  - Lectures: 28 Hours (14 Weeks X 2 Hours)
  - Tutorial: 14 hours (14 Weeks x1 Hour)
  - Practical: 28 hours (14 weeks x 2 hours)

2.1 *Prerequisites and Dependencies*

Students must have a basic understanding of the C programming language, especially the concept of arrays, functions and pointers in C.

2.2 *Objective*

The major objective of this course is to provide students with the understanding of data organization and efficient formation of complex data structures on a computer.

2.3 *Course Outcomes*

After course completion, following are the learning/course outcomes.

CO1. Understanding efficient storage mechanisms of data for easy access.
CO2. Designing and implementing various basic and advanced data structures.
CO3. Making decisions for representation of the data in the real world.
C04. Understand the features of object-oriented programming.
C05. Understand how to apply the major object-oriented concepts to implement object-oriented programs in C++, viz., encapsulation, inheritance, polymorphism etc.

3. Course Contents

Module I: Basics and Linear Data Structure (10 hours)

Algorithm, Time and space analysis of algorithms-average, best and worst case analysis, Data type-Abstract Data Type, Linear and Non linear data structure, Array, Representation of array, Stack-definition and concept, primitive operations on stack, stack as an abstract data type, implementation of push and pop operations, infix expression, prefix expression and postfix expressions, evaluating a postfix expression, converting an infix expression to postfix form, queue, primitive operations on queue, queue as ADT, implementation of queue operations, priority queue, Linked list- definition and concept, primitive operations on list, list implementation of stack, list implementation of queue, list as a data structure, list implementation of priority queue, array implementation of lists, circular list, stack as circular list, queue as circular list, Josephus problem, doubly linked list.

Module II: Non linear Data structure (5 hours)

Tree- definition and concept, Binary tree, operations on binary tree, application of binary tree, node representation of binary tree, internal and external nodes, array representation of binary tree, binary tree traversals, threaded binary tree, the Huffman algorithm, representing lists as binary tree, trees, tree traversals, general expressions as trees, evaluating an expression tree, constructing a tree.

Module III: Sorting and searching (7 hours)

Sorting, bubble sort, quick sort, selection sort, binary tree sort, heap sort, insertion sort, shell sort, merge sort, radix sort, basic search techniques, dictionary as an abstract data type, sequential search, indexed sequential search, binary search, interpolation search, binary search tree operations, efficiency of BST operations, multiway search tree, operations on MST, B tree, operations on B tree, B+ tree, digital search tree, hashing, Collision - resolution technoque

Module I V: Introduction to C++ (10 hours)

Object Oriented Programming,The origin of C++, Structure of C++ program, Creating the source file, Compiling and linking, Tokens, Keywords, Identifiers and constants, Basic data types, User-defined data types, Derived data types, Variables, Operators in C++, Scope resolution operators, Memory management operators, Type cast operators, Operator precedence, Control structures, Function ,Function prototyping, Call by reference, Inline function, Default arguments, Const arguments, Friend and virtual function, Library function, Classes and Objects, Defining member functions, Private, Public and protected member, Parameterized constructor, Static class members-static data member and static member function, Nested classes, Local classes, Passing objects to functions, Returning object, Object assignment, Arrays of objects, Pointers to objects, The this pointer, Pointer to derived type, Pointer to class members, References, Dynamic allocation operators, Constructor and Destructor, Parameterized constructor, Constructor with default argument, Execution of constructor and destructor

Module V: Function and Operator Overloading in c++ (5 hours)

Function Overloading, Overloading constructors, Copy constructors, Finding address of an overloaded function, Default function arguments, Function overloading and ambiguity, Operator overloading, Overloading nary and binary operators, Creating a member operator function, Operator overloading using a member function, Rules for overloading operators, Overloading new and delete, Overloading some special operators, Type conversions

Module VI: Inheritance in c++ (5 hours)

Defining derived class, Single inheritance, Multilevel inheritance, Multiple inheritance, Hierarchical Inheritance, Hybrid Inheritance, Constructor and destructor and inheritance

4. Laboratory Sessions (28 hours)

The practical component of this course is of one credit which amounts to 2 hours of Laboratory class per week.

- Introduction to Editors, Compilers, Debuggers and IDE's.
- Introduction to C++, Classes and Objects.
- Polymorphism.
- Inheritance.
- File I/O.
- Exception
- Link List, Operations on a Linked List (Singly).
- Stack, Queue, Tree
- Binary search tree, Multiway Search Tree.
- Linear and Binary search using Array. Sorting – Bubble sort.
- Quick sort, selection sort
- Insertion sort, Merge sort, Heap sort
-

5. Referential Sources

Useful Books and Papers

- Tanenbaum A. "Data Structures using C and C++". 3rd ed., Que Publishing, 2013.
- Lipschutz S. "Data Structures with C". 4th ed., PHI Publishing, 2015.
- Herbert Schidlt, "C++: The Complete Reference". McGraw Hill.
- Balagurusamy E., "Object Oriented Programming with C++". PHI.
- Stroustrup Bjarne, "The C++ Programming Language".

Useful Web Sources

- https://nptel.ac.in/courses/106102064/ : NP-TEL content for Data Structure.
- https://nptel.ac.in/courses/106105151/ : NP-TEL content for C++.

SWAYAM link

- https://swayam.gov.in/nd2_cec19_cs04/preview : SWAYAM course on Data Struct.
- https://swayam.gov.in/nd1_noc19_cs38/preview : SWAYAM course on C++.

1. About the Course

This is a *Core Course* that deals with designing, maintenance and transaction of database systems. This course is organized as a series of lectures with both theory and laboratory sessions. This course covers basic database concepts, data models, database architecture, relational database languages, SQL, functional dependencies and normalization, and database transactions.

2. Course Description

- Target Audience:
  - 2nd semester students of MCA programme only.
- Course Period: One semester (14 Weeks)
- Total Credit (L + T + P): 4 (2 + 1 + 1)
- Total Credit Hours: 5 hours
- Total Contact Hours: 70 hours(14 Weeks X 3 hours + 14 weeks x 2 hours)
  - Lectures: 28 Hours (14 Weeks X 2 Hours)
  - Tutorial: 14 hours (14 Weeks x1 Hour)
  - Practical: 28 hours (14 weeks x 2 hours)

2.1 *Prerequisites and Dependencies*

This is a core course of MCA programme offered in second semester. Students must have fundamental knowledge of computer including software , basic programming and discrete mathematics.

2.2 *Objective*

The main objective of the course is to equip students with the skills of database design. The principles and techniques involved in designing a productive and good database from conceptual level to implementation level are covered in this course. The course also addresses issues of database transaction and error recovery.

2.3 *Course Outcomes*

On completion of the course, students

CO1. Will have a broad understanding of database concept and DBMS software
CO2. Will be able to design a database for an application software, at conceptual level using ER modelling tool and to convert that database into implementation level using Relational model.
CO3. Will be able to model a good normalized database to remove redundant data.
CO4. Will be able to write SQL commands to work with any database.
CO5. Will have an understanding on issues involved in database transaction and error recovery.

3. Course Contents

Module I: Introduction to Databases (3 hours)

Database, characteristics of database approach, advantages of DBMS, database models, database architecture and data independence, database languages, classification of DBMSs

Module II: Entity Relationship Model (7 hours)

Database design and ER Model: overview, ER Model, Constraints, ER Diagrams, ERD Issues, weak entity sets, subclasses, superclasses, and inheritance, specialization and generalization

Module III: Relational Data Models and SQL (9 hours)

Relational model concept, relational model constraints, relational database schemas, Codd's rules, ER to relational model mapping, SQL data definition and data types, specifying constraints in SQL, retrieval queries in SQL, insert, delete and update statements in SQL, assertions, triggers, views, schema change statements.

Module IV: Relational Algebra and Calculus (7 hours)

Unary relational operations: SELECT and PROJECT, relational algebra operations from set theory, binary relational operations: JOIN and DIVISION, tuple relational calculus, domain relational calculus

Module V: Dependencies and Normal Forms (9 hours)

Importance of a good schema design, motivation for normal forms, dependency theory - functional dependencies, Armstrong's axioms for FD's, closure of a set of FD's, minimal covers, definitions of 1NF, 2NF, 3NF and BCNF, decompositions and desirable properties of them, algorithms for 3NF and BCNF normalization, multi-valued dependencies and 4NF, join dependencies and definition of 5NF.

Module VI: Transaction Processing and Error Recovery (7 hours)

concepts of transaction processing, ACID properties, concurrency control, locking based protocols for CC, error recovery and logging, undo, redo, undo-redo logging and recovery methods.

4. Laboratory Sessions (28 hours)

The practical component of this course is of one credit, i.e., 2 hours of lab classes per week. The laboratory work consists of the following.

- Introduction to DBMS software and SQL.
- Introduction to Oracle software
- DDL to create, change schema of database and relation and to grant access right.
- SQL data type.
- Insert, update, delete and retrieval queries in SQL.
- Specifying primary and foreign key and other integrity constraints.
- Nested SQL queries and joining of tables.
- Aggregate functions.
- SQL to create views.
- PL/SQL subprograms.
- Writing triggers.

5. Referential Sources

Books:

- Ramez Elmasri, Shamkant B. Navathe, Fundamentals of Database Systems.
- Avi Silberschatz, Henry Korth, S. Sudarshan, Database System Concepts.

Operating System

MCAC2404

1. About the Course

This is a *Core Course* and is aimed to make students understand the concepts of the most important system software called operating system. The course covers various issues of operating system that one needs to know to understand the structure of an operating system.

2. Course Description

- Target Audience:
  - 2nd semester students of MCA programme only.
- Course Period: One semester (14 Weeks)
- Total Credit (L + T + P): 4 (2 + 1 + 1)
- Total Credit Hours: 5 hours
- Total Contact Hours: 70 hours(14 Weeks X 3 hours + 14 weeks x 2 hours)
  - Lectures: 28 Hours (14 Weeks X 2 Hours)
  - Tutorial: 14 hours (14 Weeks x1 Hour)
  - Practical: 28 hours (14 weeks x 2 hours)

2.1 *Prerequisites and Dependencies*

Operating system course has a dependency on "Programming in C" course. A student needs to be also aware of basic concepts like hardware components, software and types of software.

2.2 *Objective*

The major objective of this course is to provide students with understanding of the overall structure of an operating system.

2.3 Course *Outcomes*

After course completion, following are the learning outcomes.

CO1. To understand the services provided by operating system and design of an operating system

CO2. Understanding Process Management module of operating system.

CO3. Understanding Memory Management module of operating system.
CO4. Understanding File System Management module of operating system.
CO5. Understanding I/O management module of operating system.
CO6. Understanding OS Security: Threats and Security Controls

3. Course Contents

Module I: Basics **(4 hours)**

Operating System Functionalities, Types of Operating Systems, Structure of Operating System, Distributed Systems, Services, System Calls, Virtual Machines, System Boot.

Module II: Process Management **(12 hours)**

Process Scheduling - Uniprocessor scheduling algorithms, Multiprocessor and Real-time scheduling algorithms, Process Synchronization - Peterson's Solution, Bakery Algorithm, Hardware Support to

Process Synchronization, Semaphores, Critical Regions, Monitors - Deadlock prevention, deadlock avoidance and Deadlock Detection and Recovery - Bankers Algorithm, Threads.

Module III: Memory Management **(11 hours)**

Swapping, Contiguous Memory Allocation, Paging Structure of the Page Table Segmentation Example: The Intel Pentium, Virtual memory, Demand Paging, Page Replacement, Thrashing, Memory Mapped Files, Allocation of Kernel Memory.

Module IV: File Systems **(5 hours)**

Contiguous, Sequential and Indexed Allocation, File system interface, File System implementation, Case study of Unix File system - Buffer Cache, Inodes, The system calls - ialloc, ifree, namei, alloc and free, Mounting and Unmounting files systems, Network File systems.

Module V: I/O System **(5 hours)**

I/O Hardware, Application I/O Interface, Kernel I/O Subsystem, Device drivers - block and character devices, streams, Character and Block device switch tables.

Module VI: Protection and Security **(5 hours)**

Goals of Protection, Principles of Protection, Domain Protection, Access Matrix, Access Control, Security Problem, Program Threats, System and Network Threats, User Authentication, Firewall.

4. Laboratory Sessions (28 hours)

This course contains 2 hours of practical classes per week. Following topics will be covered in the laboratory classes.

- Introduction to various process, memory and file management commands in windows operating system.
- Introduction to various process, memory and file management commands in linux operating system.
- Introduction to system calls in linux operating systems.
- Introduction to system calls in windows operating systems.

5. Referential Sources

Books:

- Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, “Operating System Concepts”. Wiley India Pvt. Ltd
- Maurice Bach, “Design of the Unix Operating System”. Prentice Hall India Learning Private Limited

Advanced Web Technology
MCAC3401

1. About the Course

This is a *Core Course* and is aimed at teaching skills to design interactive and dynamic web sites. The course is designed to deliver key technology components like descriptive languages and server-side program elements. In addition, the course gives specific contents that are beneficial for developing web-based solutions like communication with a relational database, data security principles and approaches. The focus of this course would be on advanced topics in emerging Web technologies. These include extensions of Web standards, combination of different Web technologies, Web toolkits and development environments, current backend Web frameworks.

2. Course Description

- Target Audience:
  - 3rd semester students of MCA programme only.
- Course Period: One semester (14 Weeks)
- Total Credit (L + T + P): 4 (2 + 1 + 1)
- Total Credit Hours: 5 hours
- Total Contact Hours: 70 hours(14 Weeks X 3 hours + 14 weeks x 2 hours)
  - Lectures: 28 Hours (14 Weeks X 2 Hours)
  - Tutorial: 14 hours (14 Weeks x1 Hour)
  - Practical: 28 hours (14 weeks x 2 hours)

2.1 *Prerequisites and Dependencies*

The course is built up on the knowledge of previous courses such as the “Basic Web Technology” course, “Data Structures using C++”.

2.2 *Objective*

The main objective of this course is to provide knowledge on web architecture, web services, server-side scripting technologies to focus on the development of web-based information systems and web services.

2.3 *Course Outcomes*

After course completion, following are the learning/course outcomes.

C01. Define the fundamental ideas and standards underlying Web Service Technology.
CO2. Define the fundamental principles for cloud applications.
CO3. Discuss concepts at the frontier of industrial practice and emerging standards.
CO4. Differentiate the major frameworks allowing to develop web services and cloud applications and assess their suitability for specific usage scenarios.

3. Course Contents

Module I: Introduction to PHP **(7 hours)**

PHP Functionalities, Datatypes, Variables, Constants, Arrays, Functions, Strings, System Calls, Explode-Implode and other native functions.

Module II: Core PHP Concepts **(20 hours)**

Handling Html Form with PHP, Working with file and Directories, Sessions and Cookies, Database connectivity using MySQL and MySQLi, Concept of PDO, Performing basic database

Operation using DML commands (such as Insert, Delete, Update, Select), Setting query parameter, Executing query Joins (Cross joins, Inner joins, Outer Joins, Self joins.), Exception Handling. Introduction to Wordpress CMS.

Module III: Introduction to Laravel **(10 hours)**

Installation, Artisan CLI, Laravel directory structure, Basic routing, Call a controller method from a route, Template inheritance, Blade conditional statements, Blade Loops, Executing PHP functions in blade, Introduction to Migrations, Migration structure, Creating a basic controller, Creating a route using a closure, Eloquent ORM Models, Eloquent ORM INSERT, READ, UPDATE, DELETE, Using models in controllers.

Module IV: Introduction to NodeJS and ExpressJS **(15 hours)**

Installation procedures, NodeJS console, REPL, TLS/SSL, Debugger, Process, Child Process, Buffers, Streams, File System, Path, Query String, Assertions, Callbacks, Events, TTY, Web-modules, Database connectivity, Intro to ExpressJS, Express.js fundamental concepts like Routing and HTTP Methods, Middleware, Cookies, REST API, Scaffolding, Templating and Error Handling.

4. Laboratory Sessions (28 hours)

This course contains 2 hours of practical classes per week. Following topics will be covered in the laboratory classes.

- Introduction to Server-side scripting technology using PHP.
- Core PHP concepts.
- Introduction to Laravel framework.
- Concept on CMS using Wordpress/Drupal.
- Introduction to NodeJS.
- Introduction to ExpressJS.

5. Referential Sources

Books:

- Papazoglou, “Web Services: Principles and Technology (2nd edition)”; ISBN: 978-0-273-73216-7, Prentice Hall, 2012
- Cerami, “Web Services Essentials”; ISBN: 0596002246, O’Reilly,2002

Useful Web Sources:

- https://www.php.net/docs.php : PHP Documentation

- https://laravel.com/docs/9.x : Laravel Documentation

Object Oriented Programming and Design using Java

MCAC3402

1. About the Course

This is a *Core Course* and is aimed at making a student comfortable with object oriented Programming and Design using Java and its features. The course is organized as a series of lectures with both theory and laboratory sessions.

2. Course Description

- Target Audience:
  - 3rd semester students of MCA programme only.
- Course Period: One semester (14 Weeks)
- Total Credit (L + T + P): 4 (2 + 1 + 1)
- Total Credit Hours: 5 hours
- Total Contact Hours: 70 hours(14 Weeks X 3 hours + 14 weeks x 2 hours)
  - Lectures: 28 Hours (14 Weeks X 2 Hours)
  - Tutorial: 14 hours (14 Weeks x1 Hour)
  - Practical: 28 hours (14 weeks x 2 hours)

2.1 *Prerequisites and Dependencies*

Students are expected to be comfortable in procedure oriented (C) and object oriented programming (C++). So “Introductory Computing using C” and “Data Structure using C++” courses are prerequisites for this course.

2.2 *Objective*

The major objective of this course is to equip students with programming skills to design high-end GUI based applications using Java APIs.

2.3 *Course Outcomes*

After course completion, following are the learning/course outcomes.

CO1. Understanding the approach to solve a problem in java.

CO2. Writing an efficient Java program with appropriate language constructs to solve a problem.

CO3. GUI based application development.

3. Course Contents

Module I: Introduction to Object Oriented Programming (2 hours)

Programming paradigm, What is Object Oriented Programming, Object oriented programming vs procedure oriented programming, Basic concepts of OOP-Encapsulation, Polymorphism and Inheritance, Object oriented language,

Module II: Core Java Programming (7 hours)

Java Overview: Genesis, Java Philosophy, Java & Internet, Object-Oriented Programming features, Java Applet and Application, Java Environment and Java Development Kit (JDK) & Java Standard Library (JSL), Java language fundamentals, The scope and lifetime of variable, Type conversion and casting, Control statements, Arrays, classes and objects: The this keyword, Garbage collection, Overloading constructor, Using object as parameters, Argument passing, Returning objects, Recursion, Introducing Access control (public, private and protected), static, final, nested classes, String class, Command-line argument

Module III: Inheritance, Exception handling (5 hours)

Inheritance: Member access and inheritance, method overriding, dynamic method dispatch, using abstract classes, using final with inheritance, the Object class; Packages, Interface, classpath, Exception handling: Fundamentals, Exception types, Java's built-in exceptions, user defined exceptions .

Module IV: Multithreading and I/O Basics (5 hours)

Multithreaded Programming: The Java thread model (thread priorities, synchronization and inter-thread communication); Deadlock, ThreadGroup, I/O Basics : (Streams, The stream classes, The predefined streams, Reading console input, writing console output, The transient and volatile modifiers, using instance of native methods

Module V: String handling, Utility classes, java.lang and java.io (7 hours)

String handling: String constructors, methods for character extraction, string searching & comparison, data conversion using valueof (), StringBuffer, Exploring java.lang: Simple type wrappers, System class, class Class, Math functions, The utility classes: Vector, Stack, HashTable, StringTokenizer, Bitset, Date, Calendar, GregorianCalendar, Random, Observable, Input/Output-Exploring java.io: The java.io classes and interface, File class and methods for creating, renaming, listing and deleting files and directories, I/O stream classes (FileInputSream, FileOutputStream, BufferedInputStream, BufferedOutputStream, PushBackInputStream, InputStreamReader, BufferedReader, BufferedWriter, PrintStream, RandomAccessFile)

Module VI: Networking, Images (5 hours)

Networking: Socket overview, Stream Sockets, Datagram sockets, Manipulating URLs, Establishing a simple Server/Client using Stream Sockets, Connectionless Client/Server Interaction with Datagrams, Images: File formats, image fundamentals, creating, loading and displaying images, ImageObserver, MediaTracker

Module VII: Applet class and Swing (6 hours)

The Applet class: applet architecture, passing parameters to applets, getDocumentBase, getCodeBase, and showDocument, AppletContext and AudioClip interfaces, Graphics class and methods for drawing lines, rectangles, polygons and ovals, Swing: Component and Container classes, Layout managers (FlowLayout, GridLayout, BorderLayout), Handling events, Adapter classes, Anonymous inner classes Swing GUI components (JLabel,

JTextField, JTextArea, JButton, JCheckBox, JRadioButton, JList, JComboBox, JScrollBar, JScrollPane, JToolTip, JPanel, JFrame), Menus: JmenuBar, JMenu, JMenuItem, JSeparator

Module VIII: JDBC **(2 hours)**

Java database connectivity (JDBC): Introduction to JDBC, type of JDBC connectivity, Establishing database connections, Accessing relational database from Java programs

Module IX: Java Beans, Java Servlets **(3 hours)**

Java Beans: Introducing JavaBeans Concepts and Bean Development kit (BDK), Using the Bean Box, Writing a simple Bean, Bean Properties (simple properties), Manipulating events in the Bean Box

4. Laboratory Sessions (28 hours)

The practical component of this course is of one credit which amounts to 2 hours of Laboratory classes per week. Following are the components that will be covered in practical sessions. **Students has to submit a minor Project using Java language at the end of this course.**

- Introduction to editor, compiler, debugger and IDE.
- Compilation process of a Java program.
- Basic java programs with input and output statements.
- Java programs to handle data types and variables.
- Java programs with branching statements.
- Java programs with looping statements.
- Java programs with array.
- Handling strings in a Java program.
- Multi-dimensional array in Java program.
- Functions in java program.
- Java programs to implement object oriented concepts.
- Multithreading concepts.
- GUI Programming (AWT/Swing).
- Working with database (JDBC/ ODBC connection).

5. Referential Sources

Books:

- Schildt Herbert, “The Complete Reference Java”. McGraw Hill.
- Balagurusamy E., “Programming with Java: A Primer”. McGraw Hill.

Useful Web Sources

- https://nptel.ac.in/courses/106105191/ : NP-TEL content.

SWAYAM Link

- https://swayam.gov.in/nd1_noc19_cs84/preview : SWAYAM course.

Introduction to Artificial Intelligence
MCAC3403

1. About the Course

This is a *Core Course*. At present, Artificial Intelligence is one of the most dominant area that studies how to realize the intelligent human behaviors on a computer. This course is organized in a series of lectures which includes both theory and tutorial sessions.

2. Course Description

- Target Audience:
  3rd semester students of MCA programme
- Course Period: One semester
- Total Credit (L + T + P): 4 (3 + 1 + 0)
- Total Credit hours: 4
- Total Contact Hours: 56 Hours (14 Weeks X 4 Hours)
  1. Lectures: 42 Hours (14 Weeks X 3 Hours)
  2. Tutorial: 14 Hours (14 Weeks x 1 Hour)
  3. Practical: Nil

2.1 *Prerequisites and Dependencies*

Students are expected to have good knowledge in algorithm, programming, graph theory for this course.

2.2 *Objective*

The main objective of this course is to provide the fundamental knowledge to the students so that they can understand what AI is. Basic principles, techniques and application of Artificial Intelligence are introduced in this course.. After completion of the course, students who become interested in AI can go for further advanced study and research.

2.3 *Course Outcomes*

After course completion, following are the learning/course outcomes.

CO1. Gain a historical perspective of AI.
CO2. Become familiar with basic principle of AI towards problem solving, Knowledge representation and planning.
CO3. Will understand different search strategies used for problem solving
CO4. Becoming familiar with learning algorithm

3. Course Contents

Module I: Overview **(10 hours)**

What is Artificial Intelligence? Turing test, history of AI, Intelligent agents, Agent programs, task environment, PEAS, Types of intelligent agent

Module II: Problem solving by searching **(16 hours)**

problem and goal formation, well defined problem and solution, searching for solution, uninformed search strategies- Breadth First Search , Uniform Cost Search, Depth First Search, Bidirectional Search, Informed search strategies-greedy best first search, A* Search, heuristic function, hill climbing search, genetic algorithm, AND-OR search tree, Adversarial search-the minimax algorithm, alpha-beta pruning, constraint satisfaction problem

Module III: Knowledge Representation **(16 hours)**

Knowledge based agents, Symbolic Logics- Introduction, Propositional Logics, Syntax and semantics of FOPL, Properties of Well Formed Formulas, Inference Rules, Knowledge representation using rules ,dealing with inconsistence- Default Reasoning and the closed world assumptions, Predicate Completion, Modal and temporal logics, Fuzzy logic and natural language computations, Probabilistic reasoning- Bayesian probabilistic reasoning, Heuristic reasoning method, Structured knowledge- Associative networks, Frame structure

Module IV: learning **(14 hours)**

Forms of learning, supervised learning, learning decision tree, unsupervised learning, artificial neural network, support vector machine, reinforcement learning

5. Referential Sources

Books:

- Dan W. Patterson," Artificial Intelligence and Expert System", Prentice Hall
- E. Rich & K. Knight, "Artificial Intelligence", Tata McGraw Hill.
- N.J. Nilson, "Principles of Artificial Intelligence", Narosa Pub. House.

Sturat J. Russell, Peter Norvig, "Artificial Intelligence ,a modern approach", Prentice Hall

Data Mining
MCAC4401

1. About the Course

This is a *Discipline Specific Elective Course* and is aimed to make students understand the concepts of data mining for extracting useful patterns, information from huge amount of data.

2. Course Description

- Target Audience:
  - 4^{th} semester students of MCA programme only.
- Course Period: One semester (14 Weeks)
- Total Credit (L + T + P): 4 (2 + 1 + 1)
- Total Credit Hours: 5 hours
- Total Contact Hours: 70 hours(14 Weeks X 3 hours + 14 weeks x 2 hours)
  - Lectures: 28 Hours (14 Weeks X 2 Hours)
  - Tutorial: 14 hours (14 Weeks x1 Hour)
  - Practical: 28 hours (14 weeks x 2 hours)

2.1 *Prerequisites and Dependencies*

Data Mining requires “Programming in C” and “Introduction to Python Programming ” as prerequisite as the algorithms involved in data processing are to be implemented using C , Python programming.

2.2 *Objective*

The major objective of this course is to provide students with understanding of Data mining and various techniques that are used to extract useful patterns from data.

2.3 *Learning Outcomes*

After course completion, following are the learning outcomes.

CO1. Understanding Data mining
CO2. Understanding Preprocessing tasks in Data analysis.
CO3. Understanding Classification techniques.
CO4. Understanding Clustering techniques.
CO5. Understanding Association Rule Mining Techniques.

3. Course Contents

Module I: Introduction (5 hours)

What is data mining, data mining tasks, types of data- attribute, measurement, data quality, data preprocessing, measure of similarity and dissimalarity

Module II: Mining Frequent Patterns, Associations, and Correlations (12 hours)

Market Basket Analysis, Frequent Itemsets, Closed Itemsets, and Association Rules, Frequent Pattern Mining, The Apriori Algorithm: Finding Frequent Itemsets Using Candidate Generation, Generating Association Rules from Frequent Itemsets, Mining Frequent Itemsets without Candidate Generation, Mining Multilevel Association Rules, Mining Multidimensional Association Rules.

Module III: Supervised Learning (15 hours)

Classification & Prediction: Decision Tree Techniques, Decision Tree Induction, Attribute Selection Measures, Tree Pruning, Scalability and Decision Tree Induction, Bayes’ Theorem, Naïve Bayesian Classification, Bayesian Belief Networks, Training Bayesian Belief Networks, Rule Extraction from a Decision Tree, A Multilayer Feed-Forward Neural Network, Defining a Network Topology, Back-Propagation Method, k-Nearest-Neighbor Classifiers, Genetic Algorithms, Regression: Linear Regression, Nonlinear Regression, Classifier Accuracy Measures, Holdout Method and Random Subsampling, Cross-validation, Bootstrap, Ensemble Methods, Bagging, Boosting.

Module IV: Unsupervised Learning (10hours)

Clustering, Types of Data, Partitioning Methods, Hierarchical Methods, Density Based Methods, Grid Based Methods, Model Based Methods, Clustering High-Dimensional Data, Constraint Based Methods, Outlier Analysis.

4. Laboratory Sessions (28 hours)

This course contains 2 hours of practical classes per week. Following topics will be covered in the laboratory classes.

1. Implementation of data preprocessing techniques
2. Implementation of Data Characterization measures
3. Implementation of Proximity Measures
4. Implementation of classification techniques
5. Implementation of Clustering Techniques
6. Using of Weka software for different data mining tasks

5. Referential Sources

Books:

- Jiawei Han and Micheline Kamber, "Data Mining: Concepts and Techniques". Morgan Kaufmann India.
- Pang-Ning Tan, Steinbach, Karpatne, Vipin Kumar, "Introduction to Data Mining", Pearson
- Ian H. Witten, Eibe Frank, Mark A. Hall, "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann

Software Engineering
MCAC4402

1. About the Course

This is a *Core Course* and it comprises of the core principles in efficient and consistent software development and maintenance. An introduction to object-oriented software development process and design has also been included in the course.

2. Course Description

- Target Audience:
  - 4th semester students of MCA programme only.
- Course Period: One semester (14 Weeks)
- Total Credit (L + T + P): 4 (2 + 1 + 1)
- Total Credit Hours: 5 hours
- Total Contact Hours: 70 hours(14 Weeks X 3 hours + 14 weeks x 2 hours)
  - Lectures: 28 Hours (14 Weeks X 2 Hours)
  - Tutorial: 14 hours (14 Weeks x1 Hour)
  - Practical: 28 hours (14 weeks x 2 hours)

2.1 *Prerequisites and Dependencies*

The students are expected to have knowledge on programming and software. In turn, this course provides foundation for Project to be carried out in the final semester of the MCA programme.

2.2 *Objective*

- Knowledge of basic Software engineering methods, practices and their appropriate application.
- Describe software engineering layered technology and Process frame work.
- A general understanding of software process models such as the waterfall and evolutionary models.
- Understanding of software requirements and the SRS documents.
- Describe data models, object models, context models and behavioural models.
- Understanding of different software architectural styles.
- Understanding of implementation issues such as modularity and coding standards.
- Understanding of approaches to verification and validation including static analysis, and reviews.
- Understanding of software testing approaches such as unit testing and integration testing.
- Understanding on quality control and how to ensure good quality software.

2.3 *Course Outcomes*

After course completion, following are the learning/course outcomes.

CO1. Define various software application domains and applying different process model used in software development.
CO2. Explain needs for software specifications.
CO3. Convert the requirements model into the design model and demonstrate use of software and user interface design principles.
CO4. Can classify different testing strategies and tactics.
CO5. Justify role of SDLC in Software Project Development.
CO6. Generate project schedule, design and develop network diagram for different type of Projects.

3. Course Contents

Module I: Introduction **(5 hours)**

Why study Software Engineering, Evolution and Impact of Software Engineering, Software Development Projects, Programs vs Products, Emergence of Software Engineering Early Computer Programming, High-Level Language Programming, Control Flow- Based Design, Data-Structure-Oriented Design, Data Flow-Oriented Design, Object-Oriented Design

Module II: Software Life Cycle Models **(6 hours)**

Use of Life Cycle Model, Classical Waterfall Model, Iterative Waterfall Model, Prototyping Waterfall Model, Evolutionary Model, Spiral Model , Comparisons of different Life Cycle Models

Module III: Software Project Management **(7 hours)**

Responsibility of a Software Project Manager, Project Planning, Matrices for Project Size Estimation: LOC, Function Point Metric, Project Estimation Technique-Empirical Estimation

Technique, Heuristic Technique, Analytical Estimation Technique, COCOMO Model, Scheduling Work Breakdown Structure, Activity Networks and Critical Path Model, Gantt Charts, PERT charts, Project Monitoring and Control, Organization Structure and Team Structure, Staffing, Risk Management- Risk Identification, Risk Assessment, Risk Containment ,Software Configuration Management

Module IV: Requirement Analysis and Specification (6 hours)

Requirement Gathering and Analysis, Software Requirement Specification, Characteristics of a good SRS Document, Functional Requirement, Traceability, Algebraic Specification

Module V: Software Design (8 hours)

Classification of Design Activities and Design Methodologies, Analysis versus Design, Cohesion and coupling, Functional Independence, Approaches to Software Design Function Oriented Design, Object-oriented Design, Function Oriented Design: Structured Analysis, Data Flow Diagrams (DFD s), Case Scenarios using DFD, Object-oriented Design: Object Modeling using UML, UML Diagrams, User Interface Design: Characteristics of a good user interface

Module VI: Coding and Testing (6 hours)

Coding Standards and Guideline, Code Review-Code Walkthrough, Code Inspection, Clean Room Testing, Software Documentation, Testing-designing test cases, Testing in the Large vs Testing in the small, Unit Testing-Driver and Stub Modules, Black-Box Testing, White Box Testing, McCabes Cyclomatic Complexity Metric, Data Flow-based Testing, Mutation Testing, Debugging, Program Analysis Tool, Integration Testing, System Testing-Performance Testing, Error Seeding ,Regression Testing

Module VII: Software Maintenance and Software Reuse (4 hours)

Characteristics of Software Maintenance, Software Reverse Engineering, estimation of Maintenance Cost, Software Reuse-basic issues in any Reuse Program, A Reuse Approach-Domain analysis, Component classification, Searching, Repository Maintenance, Reuse without Modification, Reuse at Organization level

4. Laboratory Sessions (28 hours)

The practical component of this course is of one credit which amounts to 2 hours of Laboratory class per week.

- Distinguish between Manual and Automation testing.
- Introduction of Commercial and open source automation tools.
- Understanding test automation process.
- Working with testing framework, preferably Selenium / QTP.
- Setup/Installation/configuration of testing environment, Preparation, maintenance, debugging of test-scripts, Exception handling and reporting.

5. Referential Sources

Books:

- Fundamentals of Software Engineering by Rajib Mall

- Test Automation Using selenium webdriver with java by Navneesh Garg

Useful Web Sources

- https://www.toolsqa.com/

Project
MCAC4403

1. About the Course

This is a *Core Course* and is aimed to teach student to apply the knowledge gained so far, to build some information system, application software or propose algorithm or to do some research oriented task. Students are also encouraged to do internship in any industry or organization.

2. Course Description

- Target Audience:
  - 4th semester students of MCA programme only.
- Course Period: One semester (14 Weeks)
- Total Credit (L + T + P): 4 (0 + 0 + 4)
- Total credit hours: 8
- Total Contact Hours: 112 Hours (14 Weeks X 8 Hours)
  - Lectures: Nil
  - Practical: 112 Hours (14 Weeks X 8 Hours)

2.1 *Prerequisites and Dependencies*

Prerequisites are all the courses students learned so far.

2.2 *Objective*

Primary objective is to encourage students to build some information system, application software or propose algorithm or to do some research oriented task.

2.3 *Course Outcomes*

After course completion, following are the learning/course outcomes.

CO6. Students will gain the confidence to build information system.

CO7. Students can explore different research related area of Computer Science

CO8. Students will make herself ready to work in any public or private sector organization

3. Course contents

Topics will depend on the project students worked for.

Discipline Specific Elective Courses

Graph Theory
Digital Logic and Design
Numerical Methods
Formal Language and Automata
Design and Analysis of Algorithm
Computer Graphics
Embedded Systems using Arduino
Computer Security and Cryptography
Introduction to Machine Learning
Compiler Design
Image Processing
Data Analytics with Python
Natural Language Processing
Block Chain Architecture and its use cases
Android Application Development

Graph Theory
MCAE1401

1. About the Course

This is a *Discipline Specific Elective Course*. Graph theory is the study of graphs, which are mathematical structures used to represent connections between different pair of objects in a universe of discourse. Graph theory has wide range of applications in many areas including computer science. This course is organized as a series of lectures with both theory and tutorial sessions.

2. Course Description

- Target Audience:
  1st Semester Students from MCA programme only.
- Course Period: One semester (14 Weeks)
- Total Credit (L + T + P): 4 (3+ 1 + 0)
- Total Credit Hours: 4
- Total Contact Hours: 56 Hours (14 Weeks X 4Hours)
  - Lectures: 42 Hours (14 Weeks X 3 Hours)
  - Tutorial: 14 Hours (14 Weeks x 1 hour)
  - Practical : Nil

2.1 *Prerequisites and Dependencies*

Students with good mathematical and reasoning skill can opt for the course.

2.2 *Objective*

The main objective of the course is to provide students with a broad understanding of different graph theoretic notion and its problem solving nature. Graph theoretical concepts are widely used in computer science to study and solve different problems. Different graph theoretic algorithms such as finding shortest path between pair of nodes, minimum spanning tress and graph traversal are also included in the syllabus to demonstrate students the importance of graph theory.

2.3 Course Outcomes

After completion of the course, following are the learning outcomes.

CO1. Students will understand different graph theoretic notion.
CO2. Students will be able to model and solve problem using graph theory.
CO3. Students will be able to apply different graph theoretic algorithms in applications.

3. Course Contents:

Module I: Basic Concepts of graph **(10 hours)**

Definition of Graph, Application of Graph, Incidence, Degree, Isolated and Pendant Vertex, Finite and Infinite Graphs, Null Graph, Isomorphism, Subgraphs, Walk, Paths, Circuits, Connected and Disconnected Graph, Components, Euler Graphs, Operation on Graphs,

Hamiltonian Paths and Circuits, The Travelling Salesman Problem, Cut-Sets and their properties, Cut-Vertices, Fundamental circuits, Connectivity, 1-Isomorphism and 2-Isomorphism

Module II: Trees (8 hours)
Definition of Trees, Properties of Trees, Distance and Centers, Rooted and Binary Trees, On Counting Trees, Spanning Trees, Finding all Spanning Trees of a Graph, Spanning Trees in a Weighted Graph

Module III: Planar and Dual Graphs (7 hours)
Planar Graphs, Kuratowski's Graph, Different Representation of a Planar Graph, Detection of Planarity, Geometric Dual, Combinational Dual

Module IV: Graph Representation (8 hours)
Matrix representation of Graphs, Incidence Matrix, Circuit Matrix, Fundamental Circuit Matrix and Rank, Cut-Set Matrix, Path matrix, Adjacency Matrix, Relationship between Fundamental Circuit Matrix, Fundamental Incidence Matrix and Fundamental Circuit Matrix

Module V: Coloring, Covering and Partitioning (7 hours)
Chromatic number, Chromatic Partitioning, Chromatic Polynomial, Matching, Covering, The Four Color Problem

Module VI: Directed Graph (8 hours)
Directed Graph, Types of Digraph, Directed Paths and Connectedness, Euler Digraphs, Trees with Directed Edge, Fundamental Circuits in Digraphs, Adjacency Matrix of a Digraph, Paired Comparisons and Tournaments, Acyclic Digraphs and Decyclization

Module VII: Graph Theoretic Algorithm (8 hours)
Computer Representation of Graphs, Some Basic Algorithm : Connectedness and Components, A Spanning Tree, A set of Fundamental Circuits, Cut Vertices, Shortest Path Algorithm, Depth-First Search on a Graph, Graph-Theoretic Computer Language

4. Referential Sources

Books:

- Narsingh Deo, "Graph Theory with Application to Engineering and Computer Science". PHI Publishing.
- Richard J. Trudeau, "Introduction to Graph Theory". Dover Publications.

Digital Logic and Design
MCAE1402

1. About the Course:

This is a *Discipline Specific Elective Course.* Digital logic is the basis of electronic systems. This course mainly addresses the concepts behind digital logic, its principles and technique involved in designing digital circuit. The course is organized as a series of lectures with both theory and laboratory sessions.

2. Course Description

- Target Audience:
  - 1st semester students of MCA programme only.
- Course Period: One semester (14 Weeks)
- Total Credit (L + T + P): 4 (2 + 1 + 1)
- Total Credit Hours: 5
- Total Contact Hours: 70 Hours (14 Weeks X 5 Hours)
  - Lectures: 28 Hours (14 Weeks X 2 Hours)
  - Tutorial: 14 Hours (14 Weeks x 1 Hour)
  - Practical: 28 Hours (14 weeks x 2 hours)

2.1 *Prerequisites and Dependencies*

Students are expected to have good logical and reasoning skills.

2.2 *Objective*

The course mainly deals with the concepts and principles involved in designing digital circuits. The course addresses basic concepts of a digital system including topics from number system to sequential circuits. This course will help a student in understanding how different circuits such as Flip-flop, RAM and Multiplexer of a digital system work and how to design these circuits.

2.3 *Course Outcomes*

On completion of the course, student will be able

CO1. To work with binary, octal and hexadecimal number system.

CO2. To understand the concepts behind digital system and why those systems are called so.

CO3. To understand the concept of Boolean Algebra and its application in designing digital circuits.

CO4. To design different combinational circuits using minimum number of digital gates.

CO5. To understand the concepts behind operations of flip-flops and to design sequential circuit.

3. Course Contents :

Module I: Introduction **(4 hours)**

Introduction to digital systems, Number systems like Decimal numbers, Binary numbers, Octal and Hexadecimal numbers, Number base conversions, Complements: r's complement, (r:1)'s complement, Arithmetic operations on binary numbers, Subtraction with r's and (r:1)'s complement ,Binary fixed point representation, Representation of positive and negative number, overflow.

Module II: Encoding **(2 hours)**

Information representation by code, Binary codes, BCD, Excess: 3, Alphanumeric codes: ASCII, Unicode.

Module III: Boolean algebra and logic gates **(8 hours)**

Definition of Boolean algebra: basic and axiomatic definition, Theories and properties of Boolean algebra, Boolean function:it's complement, canonical and standard form, Minterms

and Maxterms, SOP, POS, Digital logic gates: AND, OR, NOT, NAND, NOR, Exclusive OR, Exclusive NOR, Truth table, Implementation of Boolean function using gates, Universal gates : NAND and NOR implementation, Simplification of Boolean function: the Map Method, Don't care condition, the tabulation method

Module IV: Combinational Circuits (14 hours)

Combinational logic design procedure, Adder:half adder and full adder, Subtractor, Code conversion, Parity generator and parity checker, Binary parallel adder, Decimal adder, BCD adder, Magnitude comparator, Decoder, Demultiplexer, Multiplexer, Boolean function implementation, , Programmable Logic Array(PLA), Read Only Memory

Module V: Sequential Circuits (14 hours)

Sequential logic, flip:flop:RS flip:flop, JK flip:flop, D flip:flop, T flip:flop, Triggering of flip:flop, State table, State diagram, State equation, Flip:flop excitation table, Design procedure of sequential circuit, Design of counter,Synchronous and asynchronous counter, Ripple counter, BCD counter, Binary counter, Timing sequence, Johnson counter, Register, Shift register

4. Laboratory Sessions (28 hours)

The practical component of this course is of one credit of 2 hours class per week. The laboratory work consists of designing digital circuits using logic simulator tool and digital trainer kit. Following are the components that will be covered in practical sessions.

- Introduction to Integrated Circuit and IC digital logic families.
- Study the operation of different logic gates and ICs available for gates.
- Designing of different combinational circuits (half adder, full adder, parallel adder, magnitude comparator, decoder, encoder, MUX, de-MUX, parity generator etc.).
- Construction of flip-flops.
- Design sequential circuit.
- Implementation of counters (asynchronous and synchronous)

5. Referential Sources

Books:

- M.Morris Mano: Digital Logic and Computer Design, PHI (EEE)
- M.Morris Mano: Computer System Architecture, PHI (EEE)

Numerical Methods
MCAE1403

1. About the Course

This is a *Department Specific Elective Course* and is aimed at making a student comfortable on computational mathematics, by using basic algorithms underpinning computer

predictions in modern systems science. The course is organized as a series of lectures with both theory and laboratory sessions.

2. Course Description

- Target Audience:
  - 1st semester students of MCA programme only.
- Course Period: One semester (14 Weeks)
- Total Credit (L + T + P): 4 (2 + 1 + 1)
- Total Credit Hours: 5
- Total Contact Hours: 70 Hours (14 Weeks X 5 Hours)
  - Lectures: 28 Hours (14 Weeks X 2 Hours)
  - Tutorial: 14 Hours (14 Weeks x 1 Hour)
  - Practical: 28 Hours (14 weeks x 2 hours)

2.1 *Prerequisites and Dependencies*

Students are expected to be comfortable in procedural programming (C) language. So, Introductory Computing using C course is prerequisite for this course.

2.2 *Objective*

The major objective of this course is to provide the numerical methods of solving the non-linear equations, interpolation, differentiation, and integration and to improve the student's skills in numerical methods by using the numerical analysis software and computer facilities.

2.3 *Course Outcomes*

After course completion, following are the learning/course outcomes.

CO1. Demonstrate understanding of common numerical methods and how they are used to obtain approximate solutions to otherwise intractable mathematical problems.

CO2. Derive numerical methods for various mathematical operations and tasks, such as interpolation, differentiation, integration, the solution of linear and nonlinear equations, and the solution of differential equations

CO3. Analyse and evaluate the accuracy of common numerical methods.

CO4. Write efficient, well-documented C code and present numerical results in an informative way.

3. Course Contents

Module I: Solution of Equation with One variable (7 hours)

Bisection Method, Fixed-Point Iteration, Newton's Method, Error Analysis for Iterative Methods, Accelerating Convergence, Muller's Method

Module II: Interpolation (7 hours)

Interpolation and Lagrange Polynomial, Data Approximation and Neiville's Method, Divided Differences, Hermite Interpolation, Cubic Spline Interpolation, Parametric Curves.

Module III: Numerical Differentiation and Integration **(7 hours)**

Numerical Differentiation, Richardson's Extrapolation, Elements of Numerical Integration, Composite Numerical Integration, Romberg Integration, Adaptive Quadrature Methods, Gaussian Quadrature, Multiple Integration, Improper Integrals.

Module IV: Initial Value Problems for Ordinary Differential Equations **(7 hours)**

The Elementary Theory of Initial-Value-Problems, Euler's Method, Runge Kutta Method, Extrapolation Methods, Higher Order Equations and Systems of Differential Equations.

Module V: Direct Methods of Solving Linear Systems **(7 hours)**

Linear Systems of Equations, Pivoting Strategies, Linear Algebra and Matrix Inversion, Determinant of a Matrix, Matrix Factorization, Special types of Matrices.

Module VI: Solutions of Nonlinear Systems of Equations **(7 hours)**

Fixed Points for Functions of Several Variables, Newton's Method, Quasi-Newton's Method, Steepest Descent Method, Homotomy and Continuation Methods.

4. Laboratory Sessions (28 hours)

The practical component of this course is of one credit which amounts to 2 hours of Laboratory classes per week. Following are the components that will be covered in practical sessions.

- Introduction to editor, compiler, debugger and IDE.
- Writing programs with I/O statements for applying various methods discussed in theory class.

5. Referential Sources

Books:

- Richard L. Burden, "Numerical Analysis". Brooks/Cole CENGAGE Learning
- Peter Linz, Richard L. C. Wang, "Exploring Numerical Methods". Jones and Bartlett Publishers, Inc.

Useful Web Sources

- https://nptel.ac.in/courses/111106101 : NP-TEL content.

SWAYAM Link

- https://onlinecourses.swayam2.ac.in/cec20_ma11/preview : SWAYAM course.

Formal Language and Automata
MCAE1404

1. About the Course

This is a *Discipline Specific Elective Course* that deals with the study of abstract computing devices and the computational problems that can be solved using them. This course is organized as a series of lectures with theory and tutorial sessions.

2. Course Description

- Target Audience:
  - First semester students of MCA programme
- Course Period: One semester (14 Weeks)
- Total Credit (L + T + P): 4 (3 + 1 + 0)
- Total Credit Hours: 4
- Total Contact Hours: 56 Hours (14 Weeks X 4 Hours)
  - Lectures: 42 Hours (14 Weeks X 3 Hours)
  - Tutorial: 14 Hours (14 Weeks x 1 Hour)
  - Practical: Nil

2.1 *Prerequisites and Dependencies*

This is a core course offered for first semester MCA students. The students are expected to have good reasoning skill and mathematical background.

2.2 *Objective*

The main objective of the course is to give students a broad understanding of automata theory and to introduce computability theory. Automata is mainly an abstract computing device and is a useful model for many kinds of software and hardware components, used in many area of computer science such as artificial intelligence, embedded system, compiler design etc.

2.3 *Course Outcomes*

After completion of the course, a student will

CO1. Acquire a fundamental understanding of the core concepts in automata theory, formal language and grammar.
CO2. Identify formal language classes and their relationship.
CO3. Determine the decidability and intractability of computational problems.
CO4. Be able to design grammars and automata for different languages.

3. Course Contents

Module I: Finite automata (12 hours)

Finite automata , alphabets, strings, languages, deterministic finite automata, how a DFA processes a string, transition functions, the languages of a DFA, nondeterministic finite automata, the language of an NFA, Equivalence between NFA and DFA, application of NFA and DFA in text search, finite automata with epsilon transitions, uses of epsilon transitions, epsilon closure, transition function for epsilon NFA, languages of epsilon NFA, eliminating epsilon transitions

Module II: Regular expressions and languages (12 hours)

Regular expression, operators of regular expressions, building regular expressions, precedence of regular expression operators, finite automata and regular expression, converting DFA to regular expression, converting regular expressions to automata,

applications of regular expressions: lexical analysis, finding patterns in text, algebraic laws for regular expressions, pumping lemma, closure and decision properties of regular expressions, equivalence of regular languages, minimization of DFA

Module III: Context Free Grammar and Push Down Automata (14 hours)

Context free grammar, derivation using a grammar, leftmost and rightmost derivations, the language of a grammar, sentential forms, parses trees, inference, derivation and parse tree, ambiguous grammar, removing ambiguity from grammars, PDA, graphical notation of PDA, instantaneous description, acceptance by final state, acceptance by empty stack, conversion between empty stack and final state, equivalence between PDA and CFG, conversion from grammar to PDA and PDA to grammar, deterministic PDA and it's relationship with regular language, CFG and ambiguous grammar, Chomsky normal form, pumping lemma for CFL, closure and decision properties of CFL, Context sensitive language, linear bounded automata, Chomsky hierarchy.

Module IV: Turing machine (10 hours)

Turing machine, notation for Turing machine, instantaneous description for Turing machine, transition diagram for Turing machine, the language of a Turing machine, Turing machines and halting, multitrack Turing machine, nondeterministic Turing machine, simulating a Turing machine by computer and simulating a computer by Turing machine.

Module V: Undecidability and intractability (8 hours)

Languages not recursively enumerable, enumerating binary strings, codes for Turing machine, the diagonalization language, recursive languages, universal language, classes P and NP, examples of P and NP problems, polynomial time reduction.

5. Referential Sources

Books:

- Introduction to automata theory, languages and computation, John Hopcroft, Rajeev Motwani and Jeffrey Ullman
- Theory of automata, formal language and automata, S. P. Eugene Xavier
- An introduction to formal language and automata, Peter Linz

Design and Analysis of Algorithms
MCAE2401

1. About the Course

This is a *Discipline Specific Elective Course* and is aimed to introduce various types of approaches used to write an algorithm as a solution to a problem. This course is designed to enable students to analyze time requirement of an algorithm.

2. Course Description

- Target Audience:
    - 2nd Semester Students from MCA programme only.
- Course Period: One semester (14 Weeks)
- Total Credit (L + T + P): 4 (3+ 1 + 0)
- Total Credit Hours: 4
- Total Contact Hours: 56 Hours (14 Weeks X 4Hours)
    - Lectures: 42 Hours (14 Weeks X 3 Hours)
    - Tutorial: 14 Hours (14 Weeks x 1 hour)
    - Practical : Nil

2.1 *Prerequisites and Dependencies*

Since this course analyzes an algorithm, the students must be comfortable with programming and algorithm writing. So, Introductory Computing using C course is a prerequisite to take this course.

2.2 *Objective*

The major objective of this course is to classify algorithms with an intent to distinguish various approaches to solve a problem. The course also lays out a standard procedure to analyze time requirement of various types of algorithms.

2.3 *Course Outcomes*

After course completion, following are the learning/course outcomes.

CO1. Ability to analyze time requirement of an algorithm using a standard way.

CO2. Understanding various classes of algorithms that can be employed to solve a problem.

CO3. The ability to apply knowledge of computing and mathematics to algorithm design.

3. Course Contents

Module I: Introduction (10 hours)

The role of algorithms in computing, Algorithm as a technology, Growth of functions-Asymptotic notation and properties, Recurrence-the substitution method, the recursion-tree method, the master method, Randomized algorithm, Stacks and queues, Linked lists, Implementing pointers and objects, Representing rooted trees, Hash tables, Hash functions, Binary Search Trees

Module II: Some Important Algorithms (20 hours)

Sorting algorithm- Bubble sort, Selection sort, Insertion sort, Merge sort, Quick sort, Heap sort, Sorting in linear time- Radix sort, Bucket sort, Graph algorithms- Representation of Graphs, Breadth-first Search, Depth-first Search, Topological sort, Strongly connected components, Minimum spanning tree-Kruskal and Prim algorithm, Single-Source Shortest Paths- The Bellman-Ford algorithm, Dijkstra's algorithm, All-pairs Shortest Paths- The Floyd-Warshall algorithm, Number- Theoretic Algorithm- greatest common divisor, Modular arithmetic, The Chinese-remainder theorem, The RSA public-key cryptosystem

Module III: Advanced Data Structure **(6 hours)**

B-Tree, Binomial Heaps, Fibonacci Heaps, Data structure For Disjoint sets

Module IV: Advanced Design and Analysis Techniques **(20 hours)**

Dynamic Programming- Matrix-chain multiplication, Elements of dynamic programming, Longest common subsequence, Optimal binary search trees, Greedy algorithm- An activity selection problem, Elements of the greedy strategy, Huffman codes, Amortized analysis-The potential method, Dynamic table, Linear Programming- formulating problem as linear problem, the simplex algorithm, Representations of polynomials, The DFT and FET, NP-completeness problems, Approximation Algorithms-The vertex-cover problem, The travelling-salesman problem, The set-covering problem, The subset-sum problem

4. Referential Sources

Books:

- C Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, "Introduction to Algorithms". McGraw Hill.
- A. Aho, J. Hopcroft, J. Ullman, "The Design and Analysis of Algorithms". Addison-Wesley.

Computer Graphics
MCAE2402

1. About the Course

This is a *Discipline Specific Elective Course* that mainly addresses different methodologies and algorithms used in generating picture on display device. This course is organized in a series of lectures with theory, tutorial and practical sessions.

2. Course Description

- Target Audience:
  - 1st semester students of MCA programme only.
- Course Period: One semester (14 Weeks)
- Total Credit (L + T + P): 4 (2 + 1 + 1)
- Total Credit Hours: 5
- Total Contact Hours: 70 Hours (14 Weeks X 5 Hours)
  - Lectures: 28 Hours (14 Weeks X 2 Hours)
  - Tutorial: 14 Hours (14 Weeks x 1 Hour)
  - Practical: 28 Hours (14 weeks x 2 hours)

2.1 *Prerequisites and Dependencies*

Students are expected to have basic knowledge on computers including algorithm and programming. Courses on "Introductory Computing using C" and "Data Structures using C++" are prerequisite for this course.

2.2 *Objective*

The main objective of the course is to provide students a broad understanding on how a 2D or 3D image is generated on a display device. Different algorithms for generating pictures starting from a line to a 3D object is included in the course. Students will be taught to design 2D/3D graphics and apply different operations such as transformations and clipping over an image.

2.3 *Course Outcomes*

After completion of the course, students

CO1. Will have a broad idea on different algorithms to generate a 2D/3D image on a display device.
CO2. Can perform different transformations such as translation, rotation, scaling, reflection and shearing on images.
CO3. Can design 2D/3D graphics, animation and can apply clipping operations.
CO4. Can apply color models.

3. Course Contents:

Module I: Introduction (4 hours)
Overview of Graphics System-Video display devices, Raster Scan System, Random Scan System, Graphics Monitor, Hard copy devices, Graphics Software.

Module II: Output Primitives and Attributes (8 hours)
Points and lines, Line Drawing Algorithm, Circle and Ellipse Generation Algorithm, Scan line polygon fill algorithm, Boundary fill and flood fill algorithm, Antialiasing

Module III: Two Dimensional Geometric Transformations and Viewing (10 hours)
Translation, Scaling, Rotation, Reflection and Shear Transformation: Matrix representation and Homogeneous Coordinates, Composite Transformations, Transformations between coordinate system, viewing pipeline, Viewing coordinate reference frame, Window to view port coordinate transformation, Two dimensional viewing function, line and polygon clipping algorithm

Module IV: Three Dimensional Concepts (16 hours)
Three dimensional display method, Three dimensional graphics package, Three dimensional object representation-polygon surface, Curved line and surface, Quadric surface, Blobby object, Spline representation, Cubic spline interpretation method, Beizer curves and surfaces, B-spline curves and surfaces, Octrees, Three dimensional geometric modeling and transformations, Three dimensional viewing**,** visible surface detection, Surface rendering methods

Module V: Color Models and Color Application (4 hours)
Standard primaries and Chromaticity diagram, RGB color model, YIQ color model, CMY color model, HSV color model, HLS color model, Color selection and application

4. Laboratory Sessions (28 hours)

The practical component of this course is of one credit, i.e., 2 hours of classes per week. The practical component mainly contains programming the algorithms that are discussed in theory classes in any high-level language.

- Introduction to graphics package.
- How to draw pixel, line, circle etc. using graphics function and different parameters of the functions.
- Program to implement DDA algorithm.
- Program to implement Bresenham's line drawing algorithm.
- Program to implement Mid-point circle generating algorithm.
- Program to implement Mid-point ellipse drawing algorithm.
- Program to implement Scan-line polygon fill algorithm.
- Program to implement Flood fill and boundary fill algorithm.
- Draw 2D image using graphics function.
- Program for 2D geometric transformation (translation, rotation, scaling, reflection, shearing etc.).
- Program for point clipping, line clipping and polygon clipping.
- Design simple 2D animation.

5. Referential Sources

Books:

- Hearn and Baker, "Computer Graphics".
- Edward Angel, "Interactive Computer Graphics A Top-Down Approach with OpenGL". Pearson, 5th Edition, 2009.

Compiler Design
MCAE2403

1. About the Course

This is a *Discipline Specific Elective Course* that focuses on the concepts and principles in compiler design and implementation.

2. Course Description

- Target Audience:
    - 2nd Semester Students from MCA programme only.
- Course Period: One semester (14 Weeks)
- Total Credit (L + T + P): 4 (3+ 1 + 0)
- Total Credit Hours: 4
- Total Contact Hours: 56 Hours (14 Weeks X 4Hours)
    - Lectures: 42 Hours (14 Weeks X 3 Hours)
    - Tutorial: 14 Hours (14 Weeks x 1 hour)
    - Practical : Nil

2.1 *Prerequisites and Dependencies*

Students who have completed "Introductory computing using C", "Data Structure using c++", "Formal Language and Automata" courses can opt for this course.

2.2 *Objective*

The major objective of this course is to explore the principles, algorithms and data structures involved in the design and construction of compilers. Topics include context-free

grammars, lexical analysis, parsing techniques, symbol tables, error recovery, code generation and code optimization.

2.3 Course *Outcomes*

After course completion, following are the learning outcomes.

CO1. To realize basics of compiler design and its application.
CO2. To introduce different translation languages.
CO3. To understand the importance of code optimization.
CO4. To know about compiler generation tools and techniques.
CO5. To be able to design a compiler for a simple programming language.

3. Course Contents

Module I: Introduction to compiling (7 hours)

Compilers, The phase of a compiler, compiler-construction tools, A simple one pass compiler-syntax definition, Syntax-directed translation, Parsing, Lexical analysis, Incorporating a symbol table, Abstract stack machine.

Module II: Lexical Analysis (6 hours)

The role of the lexical analyzer, Input buffering, Specification of techniques, Recognition of tokens, Design of a lexical analyzer generator.

Module III: Syntax Analysis (6 hours)

The role of the parser, Context-free grammar, Top-down parsing, Bottom-up Parsing, Operator-Precedence Parsing, LR Parser, Parser Generator.

Module IV: Syntax-Directed Translation (7 hours)

Syntax-directed definition, Constructions of syntax trees, Bottom up evaluation of S-attributed definition, L-attributed definitions, Top down translation, Bottom-up evaluation of inherited attribute, Recursive evaluators, Space for attribute value at compile time, Assigning space at compiler construction time.

Module V: Type checking (6 hours)

Type systems, Specification of a simple type checker, Equivalence of type expression, Type conversions, Overloading of functions and operators, Polymorphic functions.

Module VI: Run-Time Environments (6 hours)

Source language issues, Storage organization, Storage-allocation strategies, Access to non-local names, Parameter passing, Symbol tables, Language facilities for dynamic storage allocation, Dynamic storage allocation techniques.

Module VII: Intermediate Code Generation (6 hours)

Intermediate language, Declarations, Assignment statements, Boolean expressions, Case statements, Back-patching, Procedure calls.

Module VIII: Code Generations (6 hours)

Issue in the design of a code generator, The target machine, Run-time storage management, Basic blocks and flow graphs, Register allocation and assignments, The dag representation of basic blocks, Peephole optimization, Generating code from DAGs, Dynamic programming code generation algorithm, Code-generator generators.

Module IX: Code Optimization **(6 hours)**

The principal sources of optimization, Optimization of basic blocks, Loops in flow graphs, Global data-flow analysis, Iterative solution of data-flow equations, Code improving transformations, Dealing with aliases, Data-flow analysis of structured flow graph, A tool for data-flow analysis, Estimation of types, Symbolic debugging of optimized code.

4. Referential Sources

Books :

- Aho Alfred V. and Ullman J., "Principles of Compiler Design". Addison-Wesley.
- Aho Alfred V., Lam Monica S., Sethi Ravi, and Ullman Jeffrey D., "Compilers: Principles, Techniques, and Tools".
- Grune Dick, Bal Henri E., Cerial J.H. Jacobs and Langendoen K., "Modern Compiler Design". Springer

Useful Web Sources

- https://nptel.ac.in/courses/106105190/ **: NP-TEL content.**

Embedded Systems
MCAE2404

1. About the Course

This is a *Discipline Specific Elective Course* and is aimed at introducing students with concepts of Embedded programming, Microcontroller, Microprocessor, KEIL language for 8051 devices.

2. Course Description

- Target Audience:
  - 2nd semester students of MCA programme only.
- Course Period: One semester (14 Weeks)
- Total Credit (L + T + P): 4 (2 + 1 + 1)
- Total Credit Hours: 5
- Total Contact Hours: 70 Hours (14 Weeks X 5 Hours)
  - Lectures: 28 Hours (14 Weeks X 2 Hours)
  - Tutorial: 14 Hours (14 Weeks x 1 Hour)
  - Practical: 28 Hours (14 weeks x 2 hours)

2.1 *Prerequisites and Dependencies*

Students are expected to have basic knowledge on computers including algorithm and programming. Courses on “Introductory Computing using C” and “Data Structures using C++” are prerequisite for this course.

2.2 *Objective*

This course is designed for students to educate in various Embedded Development strategies and to introduce Bus Communication in processors, Input/output interfacing by imparting knowledge on various processor scheduling algorithms.

2.3 *Course Outcomes*

After course completion, following are the learning/course outcomes.

- The students will be specialized in Embedded System Design
- Students will learn open-source computer hardware/software platform for building digital interactive devices

3. Course Contents

Module I: Introduction to Embedded System Design (5 hours)

Definition of Embedded System, Embedded System Vs General Computing Systems, History, Classification based on generation, complexity etc. Major application areas. Purposes/specific features, recent trends.

Module II: Embedded System Architecture Design (15 hours)

Hardware architecture, its different components with functionality. Different types of processors used their trade-offs features Examples of Domain specific embedded systems with examples e.g. working of Washing machine, automobile etc. Networking concept in embedded system Different buses used I2C PCI CANetc. Software architecture, Embedded operating system architecture categories of embedded operating system, Application software options with high level and assembly level language and different tools used for software development. Process of creation of ROM image/firmware design Study of some microcontroller/processor 8051 / PIC /AVR /ARM/DSP study of Embedded readymade boards like Arduino, Raspberry Pi, implementing small projects.

Module III: Design (12 hours)

Process of Embedded System Development, Different models, waterfall model, requirement analysis, design tradeoffs, hardware software co design different hardware platforms - single board PC add on cards custom made hardware platforms. communication interfaces RS232 RS422, USB, Infrared, IEEE 1394 firmware Ethernet, IEEE 802.11 Bluetooth Embedded firmware design, creation of ROM image.

Module IV: Programming (5 hours)

Different programming options Assembly High level for Embedded systems. Requirement of Embedded real time Operating Systems its features implementation

Module V: Development and Testing (5 hours)

Testing of Embedded systems, Embedded product development life cycle EDLC and its importance, Latest trends in Embedded industry, Fundamental concept in RT Linux and Navigation Systems

4. Laboratory Session (28 hours)

The practical component of this course is of one credit, i.e., 2 hours of classes per week. The practical component mainly contains circuit design using Proteus PCB Simulator or any other simulating software and writing C programs with Keil language.

- Introduction to Software/Simulation/Compiler/IDE
- Intro to C Programming with Keil language

5. Referential Sources

Books:

- K.V. Shibu, "Introduction to Embedded Systems", Mc Graw Hill Education
- Raj Kamal, "Embedded Systems Architecture programming and design", Tata Mc Graw Hill
- K.V.K. Prasad, "Embedded Real Time Systems concept design and programming", Dreamtech
- Mazidi and Mazidi, "8051 microcontrollers and embedded Systems".

Computer Security and Cryptography MCAE3401

1. About the Course

This is a *Discipline Specific Elective Course* and is aimed at making a student aware of various security issues encountered while accessing software and websites. The course is organized as a series of lectures with both theory and tutorial sessions.

2. Course Description

- Target Audience: 3rd semester students of MCA programme
- Course Period: One semester
- Total Credit (L + T + P): 4 (3 + 1 + 0)
- Total Credit hours: 4
- Total Contact Hours: 56 Hours (14 Weeks X 4 Hours)
  - Lectures: 42 Hours (14 Weeks X 3 Hours)
  - Tutorial: 14 Hours (14 Weeks x 1 Hour)
  - Practical: Nil

2.1 *Prerequisites and Dependencies*

There is no prerequisite for this course.

2.2 *Objective*

The major objective of this course is to equip students with awareness of various threats that are encountered while accessing software and web services

2.3 *Course Outcomes*

After course completion, following are the learning/course outcomes.

CO1. Explain the fundamentals of cryptography, such as encryption, digital signatures and secure hashes
CO2. Select appropriate techniques and apply them to solve a given problem
CO3. Design and evaluate security protocols appropriate for a given situation
CO4. Demonstrate an understanding of the mathematical underpinning of cryptography
CO5. Demonstrate an understanding of some legal and socio-ethical issues surrounding cryptography
CO6. To understand various protocols for network security to protect against the threats in the networks.

3. Course Contents

Module I: Introduction to Cryptography and Block Ciphers (8 hours)

Introduction to security attacks - services and mechanism - introduction to cryptography - Conventional Encryption: Conventional encryption model - classical encryption techniques - substitution ciphers and transposition ciphers – cryptanalysis – steganography - stream and blockciphers - Modern Block Ciphers: Block ciphers principals - Shannon's theory of confusion and diffusion - fiestal structure - data encryption standard (DES) - strength of DES - differential and linearcrypt analysis of DES - block cipher modes of operations - triple DES – AES.

Module II: Confidentiality and Modular Arithmetic (10 hours)

Confidentiality using conventional encryption - traffic confidentiality - key distribution - random number generation - Introduction to graph - ring and field - prime and relative prime numbers - modular arithmetic - Fermat's and Euler's theorem - primality testing - Euclid's Algorithm - Chinese Remainder theorem - discrete algorithms.

Module III: Public key cryptography and Authentication requirements (10 hours)

Principles of public key crypto systems - RSA algorithm - security of RSA - key management – Diffle-Hellman key exchange algorithm - introductory idea of Elliptic curve cryptography – Elgamel encryption - Message Authentication and Hash Function: Authentication requirements - authentication functions - message authentication code - hash functions - birthday attacks – security of hash functions and MACS.

Module IV: Integrity checks and Authentication algorithms (10 hours)

MD5 message digest algorithm - Secure hash algorithm (SHA) Digital Signatures: Digital Signatures - authentication protocols - digital signature standards (DSS) - proof of digital signature algorithm - Authentication Applications: Kerberos and X.509 - directory authentication service - electronic mail security-pretty good privacy (PGP) - S/MIME.

Module V: IP Security and Key Management **(8 hours)**

IP Security: Architecture - Authentication header - Encapsulating security payloads - combining security associations - key management.

Module VI: Web and System Security **(10 hours)**

Web Security: Secure socket layer and transport layer security - secure electronic transaction
(SET) - System Security: Intruders - Viruses and related threads - firewall design principals – trusted systems.

4. Referential Sources

Books:

- William Stallings, "Cryptography and Network security Principles and Practices", Pearson/PHI.
- Wade Trappe, Lawrence C Washington, "Introduction to Cryptography with coding theory", Pearson.
- W. Mao, "Modern Cryptography – Theory and Practice", Pearson Education.

Useful Web Sources:

- https://nptel.ac.in/courses/106105031 : Cryptography and Network Security

Introduction to Machine Learning
MCAE3402

1. About the Course

This is a *Discipline Specific Elective Course.* It is an introductory level PG course and is aimed to make students familiar with the key algorithms and theory that form the core of machine learning.

2. Course Description

- Target Audience:
  - 3rd semester students of MCA programme only.
- Course Period: One semester (14 Weeks)
- Total Credit (L + T + P): 4 (2 + 1 + 1)
- Total Credit Hours: 5
- Total Contact Hours: 70 Hours (14 Weeks X 5 Hours)
  - Lectures: 28 Hours (14 Weeks X 2 Hours)
  - Tutorial: 14 Hours (14 Weeks x 1 Hour)
  - Practical: 28 Hours (14 weeks x 2 hours)

2.1 *Prerequisites and Dependencies*

Students are expected to have basic knowledge on probability, statistics, algorithm and computer programming for registration in this course. Therefore "Discrete Mathematics", "Introductory computing using C" are considered to be prerequisite courses. For practical session, students are expected to have knowledge on python programming.

2.2 *Objective*

The primary objective of this course is to give students a basic understanding to machine learning and to study and construct of computer algorithms that improves automatically through experience.

2.3 *Course Outcomes*

After course completion, following are the learning/course outcomes.

- CO1. Understand a wide variety of learning algorithms
- CO2. Understand how to evaluate model generated from data
- CO3. Understand the difference between supervised and unsupervised learning
- CO4. To develop skills of using recent machine learning software tools to evaluate learning algorithms and model selection for solving practical problems

3. Course Contents

Module I: Introduction (4 hours)

Well-Posed Learning Problems, Definition of learning system, Designing a learning system-training data, concept representation, function approximation, Perspective and issues in machine learning, supervised and unsupervised learning

Module II: Concept Learning (5 hours)

Concept learning task, Concept learning as search-general to specific ordering of hypothesis, Finding a Maximally Specific Hypothesis, Version space and the candidate -elimination algorithm, Inductive bias

Module III: Decision Tree Learning (8 hours)

Introduction, Decision tree representation, Decision tree learning algorithm, Hypothesis Space Search in Decision Tree learning, Inductive bias in decision tree learning, Issues in decision tree learning, Entropy Based Node selection, ID3 Algorithm, Random Forest

Module IV: Artificial Neural Network (7 hours)

Introduction, Neural Network representation, perceptrons, Multilayer and backpropagation algorithms, Convolutional network, Recurrent network

Module V: Bayesian Learning (9 hours)

Introduction, Bayes Theorem, Bayes theorem and concept learning, Maximum likelihood and least -square error hypothesis, MAP Hypothesis, Minimum Description Length (MDL) principle, Bayesian Classifiers Bayes optimal classifier, Naive Bayes classifier, Bayes optimal classifier

Module VI: Linear model and Support Vector Machine (9 hours)

Linear models for classification, Discriminant Functions, Probabilistic Generative Classifiers, Probabilistic Discriminative Classifiers, Linear models for Regression ,Linear basis function models , Bayesian linear regression , Bias-variance decomposition, Theory of SVM, VC dimension, Linearly separable data , Non-linearly separable data

4. Laboratory Sessions (28 hours)

This course contains 2 hours of practical classes per week. Following topics will be covered in the laboratory classes.

1. Logistic Regression classifier
2. Linear Regression and Gradient Descent
3 .Decision Tree
4. Single layer Backpropagation
5. SVM

5. Referential Sources

Books:

- Tom M. Mitchell,"Machine learning",McGraw Hill
- O Theobald, "Machine Learning for Absolute Beginners: A Plain English Introduction", Scatterplot Press
- D. Barber," Bayseian Reasoning and machine learning", 2012
- S. Rogers and M. Girolami, "A first course in Machine Learning", CRC Press, 2011

Advanced Operating System
MCAE3403

1. About the Course

This is a *Discipline Specific Elective Course* and is aimed to introduce students about advance concepts on Operating System. One module is totally dedicated for distributed systems. Case studies are also introduced to covers issues of design and implementation of modern operating systems.

2. Course Description

- Target Audience:
  - 3rd Semester Students from MCA programme only.
- Course Period: One semester (14 Weeks)
- Total Credit (L + T + P): 4 (3+ 1 + 0)
- Total Credit Hours: 4
- Total Contact Hours: 56 Hours (14 Weeks X 4Hours)
  - Lectures: 42 Hours (14 Weeks X 3 Hours)
  - Tutorial: 14 Hours (14 Weeks x 1 hour)
  - Practical : Nil

2.1 *Prerequisites and Dependencies*

"Operating System" course is prerequisite.

2.2 *Objective*

Main objective of this course is to introduce students to some advanced concepts on operating system such as distributed system, Real time and multimedia system etc. Case studies on some operating systems are also included in this course to realize students about design and implementation of sytem.

2.3 *Course Outcome*

After completion of this course, students will be able to

CO1: understand different types of system calls involved in process control, file management, device management, information management, communication and protection

CO2: Understand operating system structure.

CO3: to understand distributed system, design issue, distributes file system and synchronization in distributed system

CO4: to understand design issue of real time and multimedia system.

CO5: to understand design and implementation of modern OS such as Linux, Windows erc.

CO6. Knowing the state-of-the-art research on operating systems

3. Course Contents

Module I: Operating System Structure **(11 hours)**

Operating System services, system calls, Types of system call-process control, file management, device management, information management, communication and protection, system programs, operating system design and implementation, operating system structure-simple, layered, microkernels and modules, virtual machines, operating system debugging, Perfomance tuning, Dtrace

Module II: Distributed System **(20 hours)**

Introduction to Distributed Systems, Distributed system models, Design issues in DS , Communication in Distributed System-Inter process communication: Message passing model, Remote procedure call and implementation issues, Point to point and Group communication , Client Server model & its implementation, Socket programming, Processes and processors in distributed systems-Threads, system model, processor allocation, scheduling in distributed systems: Load balancing and sharing approach, fault tolerance, Real time distributed systems, Process migration and related issues, Distributed File Systems-Introduction, features & goal of distributed file system, file models, file accessing models, file sharing semantics, file caching scheme, file replication, fault tolerance, trends in distributed file system, Distributed synchronization

Module III: Special Purpose Systems **(15 hours)**

Real time systems-overview, System characteristics, Features of Real-Time Kernels, Implementation of Real-Time Operating systems, Real-Time CPU Scheduling

Multimedia systems- multimedia, requirement of multimedia kernels, CPU scheduling, Disk Scheduling, Network management

Module IV: Case Studies **(10 hours)**

The Linux System, Windows OS, History of Operating system-some influential early OSsuch as Atlas MULTICS, IBM OS/360 etc

4. Referential Sources

Books:

- Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, “Operating System Concepts”. Wiley India Pvt. Ltd
- Maurice Bach, “Design of the Unix Operating System”. Prentice Hall India Learning Private Limited
- Andrew S. Tanenbaum, Marteen Van Steen, “Distributed Systems: Principles and Paradigms”, Pearson Education

Image Processing
MCAE3404

1. About the Course

This is a *Discipline Specific Elective Course* and is aimed at presenting foundation and advanced concepts of image processing.

2. Course Description

- Target Audience:
  - 3rd Semester Students from MCA programme only.
- Course Period: One semester (14 Weeks)
- Total Credit (L + T + P): 4 (3+ 1 + 0)
- Total Credit Hours: 4
- Total Contact Hours: 56 Hours (14 Weeks X 4Hours)
  - Lectures: 42 Hours (14 Weeks X 3 Hours)
  - Tutorial: 14 Hours (14 Weeks x 1 hour)
  - Practical : Nil

2.1 *Prerequisites and Dependencies*

“Computer Graphics” and “Data Structure using Object Oriented Programming in C++” courses are prerequisites. The students are expected to have elementary knowledge of basic mathematics.

2.2 *Objective*

The course aims to introduce the fundamental and analytical techniques to be employed in image processing, as well as the concept of image processing.

2.3 *Course Outcomes*

At the completion of this course, the students should be able to do the following:

CO1: Understand the need for image transformation and its properties for different types of image transformation.

CO2: Development of image processing applications.

CO3: Understand the rapid advances in image processing.

CO4: Learn about the various techniques used to enhance images.

CO5: Learn various causes of image degradation and give an overview of image restoration techniques.

CO6: understands need for image compression and to learn spatial and frequency domain techniques of image compression.

CO7: Learn about various feature extraction techniques for image analysis and recognition

3. Course Contents

Module I: Introduction (7 hours)

Examples of fields that use digital image processing, fundamental steps in digital image processing, components of image processing system., A simple image formation model, image sampling and quantization, basic relationships between pixels.

Module II: Image enhancement in the spatial domain (9 hours)

Basic gray-level transformation, histogram processing, enhancement using arithmetic and logic operators, basic spatial filtering, smoothing and sharpening spatial filters, combining the spatial enhancement methods.

Module III: Image restoration (8 hours)

A model of the image degradation/restoration process, noise models, restoration in the presence of noise only spatial filtering, Weiner filtering, constrained least squares filtering, geometric transforms; Introduction to the Fourier transform and the frequency domain, estimating the degradation function.

Module IV: Color Image Processing (8 hours)

Color fundamentals, color models, pseudo color image processing, basics of full color image processing, color transforms, smoothing and sharpening, color segmentation

Module V: Image Compression (8 hours)

Fundamentals, image compression models, error-free compression, lossy predictive coding, image compression standards

Module VI: Morphological Image Processing (8 hours)

Preliminaries, dilation, erosion, open and closing, hit or miss transformation, basic morphologic algorithms

Module VII: Object Recognition (8 hours)

Patterns and patterns classes, recognition based on decision theoretic methods, matching, optimum statistical classifiers, neural networks, structural methods matching shape numbers, string matching

4. Referential Sources

Books :

- Rafael C. Gonzalez, “Digital Image Processing”.

Useful Web Sources

- https://nptel.ac.in/courses/117105079/ : NP-TEL content.

Data Analytics with Python
MCAE4401

1. About the Course

This is a *Discipline Specific Elective Course* and is aimed to teach students how to analyze data using Python language.

2. Course Description

- Target Audience:
  - 4th semester students of MCA programme only.
- Course Period: One semester (14 Weeks)
- Total Credit (L + T + P): 4 (2 + 1 + 1)
- Total Credit Hours: 5
- Total Contact Hours: 70 Hours (14 Weeks X 5 Hours)
  - Lectures: 28 Hours (14 Weeks X 2 Hours)
  - Tutorial: 14 Hours (14 Weeks x 1 Hour)
  - Practical: 28 Hours (14 weeks x 2 hours)

2.1 *Prerequisites and Dependencies*

“Introduction to Python Programming” and “Data Mining” are the prerequisite courses.

2.2 *Objective*

The main objective of this course is to teach students how to prepare data for analysis, perform analysis, create meaning data visualization and extract some meaning information from data or predict future trends etc.

2.3 *Course Outcomes*

After course completion, following are the learning/course outcomes.

- CO1. How to import Dataset and clean the data
- CO2. How to manipulate dataframe
- CO3. Summarize the data
- CO4. Visualize the data
- CO5. How to bulid and evaluate model

3. Course Contents

Module I: Python Fundamentals **(12 hours)**

Data structure, control statements, functions, object and classes, exception handling, file handling, Python Package for Data Analysis, Working with Numpy and Punda

Module II: Data preprocessing **(10 hours)**

Dataset and different type of attributes, Data quality, Noise, understanding domain and dataset, basic insights from data set, Importance of data preprocessing, Structured and unstructured data, Importing and exporting data, cleaning and preparing of data, Identify and handling of missing value, data formatting and data normalization

Module III: Processing, Summarizing , Visualization **(10 hours)**

Descriptive statistics, grouping, ANOVA, Corelation, Basic of data visualization and data visualization tools, Seaborn creating and plotting maps,

Module IV: Model Development and evaluation **(10 hours)**

Linear regression, model evaluation using visualization, R-Squared and MSE for In-Sample evaluation, Prediction and decision making, over-fitting, under-fitting and model selection,

4. Laboratory Sessions (28 hours)

The practical component of this course is of two credit which amounts to 4 hours of Laboratory classes per week.

- Writing program in python
- Working with Numpy and Panda Package
- Importing and exporting data set
- Cleaning dataset and handling with missing value
- Visualizing data using Matplotlib
- Scikit-learn installation and working with it

5. Referential Sources

Books:

- Wes McKinney, “Python for Data Analysis- data wrangling with Pandas, NumPy and Ipython”, O’REILLY
- Fabio Nelli, “Python Data Analytics- with Pandas, NumPy and Matplotlib”, Apress

Natural Language Processing

MCAE4402

1. About the Course

This is a *Discipline Specific Elective Course.* This course is about how to program computers to process and analyze large amount of natural language data.

2. Course Description

- Target Audience:
  - 4th Semester Students from MCA programme only.
- Course Period: One semester (14 Weeks)
- Total Credit (L + T + P): 4 (3+ 1 + 0)
- Total Credit Hours: 4
- Total Contact Hours: 56 Hours (14 Weeks X 4Hours)
  - Lectures: 42 Hours (14 Weeks X 3 Hours)
  - Tutorial: 14 Hours (14 Weeks x 1 hour)
  - Practical : Nil

2.1 *Prerequisites and Dependencies*

Students are expected to have Knowledge on Formal Language and Automata Theory.

2.2 *Objective*

The area of natural language processing (NLP) is expanding quickly and has broad applications in the humanities, social sciences, and hard sciences. Its capacity is to efficiently collect, use, and evaluate linguistic and textual data. This course serves as an introduction to natural language processing's most popular and practical current approaches, tactics, and toolkits from a theoretical and methodological standpoint.

2.3 *Course Outcomes*

After course completion, following are the learning/course outcomes.

CO1. Understand language and the tools that are available to efficiently study and analyze large collections of text.

CO2. Analyze and discuss the effects of electronic communication on our language

CO3. Learn natural language processing with manual and automated approaches.

CO4. Learn computational frameworks for natural language processing.

3. Course Contents

Module I: **(12 hours)**

Introduction A computational framework for natural language, description of English or an Indian language in the frame work, lexicon, algorithms and data structures for implementation of the framework, Finite state automata, the different analysis levels used for NLP (morphological, syntactic, semantic, pragmatic, Recursive and augmented transition networks. Applications like machine translations.

Module II: **(12 hours)**

Word level and syntactic analysis Word Level Analysis: Regular Expressions, Finite-State Automata, Morphological Parsing, Spelling Error Detection and correction, Words and Word classes, Part-of Speech Tagging. Syntactic Analysis: Context-free Grammar, Constituency,

Parsing-Probabilistic Parsing. Machine readable dictionaries and lexical databases, RTN, ATN.

Module III: **(12 hours)**

Semantic analysis Semantic Analysis: Meaning Representation, Lexical Semantics, Ambiguity, Word Sense Disambiguation. Discourse Processing: cohesion, Reference Resolution, Discourse Coherence and Structure. Knowledge Representation, reasoning.

Module IV: **(12 hours)**

Natural language generation Natural Language Generation (NLG): Architecture of NLG Systems, Generation Tasks and Representations, Application of NLG. Machine Translation: Problems in Machine Translation, Characteristics of Indian Languages, Machine Translation Approaches, Translation involving Indian Languages.

Module V: **(8 hours)**

Information retrieval and lexical resources Information Retrieval: Design features of Information Retrieval Systems, Classical, Non-classical, Alternative Models of Information Retrieval, valuation Lexical Resources: World Net, Frame Net, Stemmers, POS Tagger

5. Referential Sources

Books:

- Natural Language understanding by James Allen, Pearson Education, 2002.
- NLP: A Paninian Perspective by Akshar Bharati, Vineet Chaitanya, and Rajeev Sangal, Prentice Hall, 2016.
- Meaning and Grammar by G. Chirchia and S. McConnell Ginet, MIT Press, 1990.
- An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition by Daniel Jurafsky and James H. Martin, Pearson Education, 2006.
- Natural language processing in Prolog by Gazdar, & Mellish, Addison-Wesley

Block Chain Architecture and its use cases
MCAE4403

1. About the Course

This is a *Discipline Specific Elective Course* and is aimed at students who wants to learn a new and emerging technology in the field of computer science. Blockchain saw tremendous

growth in the past few years, but there is still a gap present in Enterprise adoption of Blockchain. This gap arises due to differences between the Blockchain solution from the traditional software solution. This course has been designed in such a way that one can get a thorough understanding of how to design Enterprise Architecture with regards to different business and technology considerations.

2. Course Description

- Target Audience:
  - 4th Semester Students from MCA programme only.
- Course Period: One semester (14 Weeks)
- Total Credit (L + T + P): 4 (3+ 1 + 0)
- Total Credit Hours: 4
- Total Contact Hours: 56 Hours (14 Weeks X 4Hours)
  - Lectures: 42 Hours (14 Weeks X 3 Hours)
  - Tutorial: 14 Hours (14 Weeks x 1 hour)
  - Practical : Nil

2.1 *Prerequisites and Dependencies*

Having Knowledge about Software Architecture, Network Security, Design Patterns, Data Mining and Python programming will help in gaining much more from this course. The medium of communication in the class is English, so students have to have communication, reading and apprehension skills of English.

2.2 *Objective*

The objective of this course is to provide conceptual understanding of block chain technology and how it can be used in Industry. The course covers the technological underpinning of block Chain operations in both theoretical and practical implementation of solutions using Ethereum.

2.3 *Course Outcomes*

After course completion, following are the learning/course outcomes.

CO1. Understanding block chain technology
CO2. Understand Cryptocurrency
CO3. Understand Smart contract
CO4. Use Remix IDE
CO5. Develop block-chain based solutions and write smart contract using Ethereum Framework.
CO6. Deploy Decentralized Application

3. Course Contents

Module I: Introduction **(7 hours)**

Introduction: Overview of Block chain, History of Blockchain, Peer to Peer Network, Smart Contract, Wallet , Digital Currency, Ledgers, Types of Blockchain Platform.

Module II: Consensus Mechanism **(9 hours)**

Permissioned Blockchain, Permissionless Blockchain , Different Consensus Mechanism- Proof of Work, Proof of Stake, Proof of Activity, Proof of Burn, Proof of Elapsed Time, Proof of Authority, Proof of Importance.

Module III: Cryptocurrency and Wallet **(10 hours)**

Types of Wallet, Desktop Wallet, App based Wallet, Browser based wallet, Metamask, Creating a account in Metamask, Use of faucet to fund wallet, transfer of cryptocurrency in metamask.

Module IV: Smart Contract and Ethereum **(10 hours)**

Overview of Ethereum, Writing Smart Contract in Solidity, Remix IDE , Different networks of ethereum, understanding blocks practically at blockhcain.com, how to compile and deploy smart contract in remix.

Module V: Understanding Hyperledger Fabric **(10 hours)**

Overview of Open source Hyperledger project, Hyperledger Fabric- Architecture, Identities and Policies, Membership and Access Control, Channels, Transaction Validation, Writing smart contract using Hyperledger Fabric.

Module VI: Use Cases **(10 hours)**

Enterprise application of Block chain: Cross border payments, Know Your Customer (KYC), Food Security, Block chain enabled Trade, We Trade – Trade Finance Network, Supply Chain Financing, Identity on Block chain, Blockchain in energy sector, Blockchain in governance.

4. Referential Sources

Books:

- Melanie Swan, “Blockchain: Blueprint for a New Economy”.
- Imran Bashier, “Mastering Blockchain: Deeper insights into decentralization, cryptography, Bitcoin, and popular Blockchain frameworks”
- Andreas M. Antonopoulos, “Mastering Ethereum: Building Smart Contracts and DApps”. O’Reilly Publications

SWAYAM Link:

- https://onlinecourses.swayam2.ac.in/aic21_ge01/preview : Blockchain Architecture

Android Application Development
MCAE4404

1. About the Course

This is an *Elective Course* and is aimed to make students capable of designing Android apps.

2. Course Description

- Target Audience:
    - 4th semester students of MCA programme only.
- Course Period: One semester (14 Weeks)
- Total Credit (L + T + P): 4 (2 + 1 + 1)
- Total Credit Hours: 5
- Total Contact Hours: 70 Hours (14 Weeks X 5 Hours)
    - Lectures: 28 Hours (14 Weeks X 2 Hours)
    - Tutorial: 14 Hours (14 Weeks x 1 Hour)
    - Practical: 28 Hours (14 weeks x 2 hours)

2.1 *Prerequisites and Dependencies*

"Object Oriented Programming and Design using Java " is a prerequisite course.

2.2 *Objective*

The major objective of this course is to provide students the skills of designing Android apps.

2.3 *Learning Outcomes*

After course completion, following are the learning outcomes.

- Understanding Architecture of Android OS.
- Ability to design application software for Android operating system.

3. Course Contents

Module I: Introduction (10 hours)

Brief history of mobile applications, Different types of mobile application, impact of mobile applications, Overview of the Android system: Kernel, HAL, Android system services, Binder IPC, Application framework, Recovery, Virtual hardware reference platforms: Introduction to the x86-based Android emulator, Introduction to ranchu,, VirtualBox-based Android emulators, Setting up development environment, Dalvik Virtual Machine & .apk file extension, Basic Building blocks - Activities, Services, Broadcast Receivers & Content providers, UI Components - Views & notifications, Components for communication -Intents & Intent Filters, Android API levels (versions & version names)

Module II: Application Structure (6 hours)

AndroidManifest.xml, Uses-permission & uses-SDK, Resources & R.java, Assets, Layouts & Drawable Resources, Activities and Activity lifecycle, First sample Application.

Module III: Emulator (10 hours)

Launching emulator, Editing emulator settings, Emulator shortcuts, Logcat usage, Introduction to DDMS, Hello World App, Creating your first project, The manifest file, Layout resource, Running app on Emulator, Second App: Develop an app for demonstrating the communication between Intents.

Module IV: Design Components **(10 hours)**

Form widgets, Text Fields, Layouts, Option menu, Context menu, Sub menu, menu from xml, menu via code, Explicit Intents, Implicit intents, Time and Date, Images and media, Composite, Alert Dialogs & Toast, Popup

Module V: Content Providers **(6 hours)**

SQLite Programming, SQLite Open Helper, SQLite Database, Cursor, Reading and updating Contacts.

4. Laboratory Sessions (28 hours)

This course contains 2 hours of practical classes per week. Following topics will be covered in the laboratory classes.

- Introduction to the emulator.
- Writing simple programs.
- Writing programs with UI components.
- Writing programs with content providers.

5. Referential Sources

Books:

- Ye Roger, “Android System Programming”. Packt Publishing Limited.
- Erik Hellman, “Android Programming”. Wiley India.

Skill Enhancement Courses

Basic Web Technology

Introduction to Latex and R Programming

Basic Web Technology
MCAS1201

1. About the Course

This is a *Skill Enhancement Course* and is aimed at teaching skills to design interactive websites. The course is designed to deliver key technology components like descriptive language and client-side program elements.

2. Course Description

- Target Audience: 1st semester student of MCA programme
- Total Credit (L + T + P): 2 (1 + 0 + 1)
- Total Credit Hours: 3
- Course Period: One semester (14 Weeks)
    - Total Contact Hours: 42 Hours (14 Weeks X 3 Hours)
    - Lectures: 14 Hours (14 Weeks X 1 Hour)
    - Practical: 28 Hours (14 Weeks X 2 Hours)

2.1 *Prerequisites and Dependencies*

The course does not have any prerequisite.

2.2 *Objective*

The main objective of this course is to provide knowledge on web architecture, web services and client-side technologies to focus on the development of web sites.

2.3 *Course Outcomes*

After course completion, following are the learning/course outcomes.

CO1. Understanding fundamental web concepts (HTTP, URIs, Web browsers, etc.).
CO2. Acquiring skills to write Client-side scripts.
CO3. Students will acquire knowledge on HTML, CSS, XML and JavaScript

3. Course Contents

Module I: Introduction **(3 hours)**

Computer and network, Intranet vs Internet, Client-Server Computing, IP address, Internet services, Hyper Text Transfer Protocol(HTTP), HTTP transaction-persistent vs non-persistent , Ports and sockets, Proxy Server.

Module II: World Wide Web **(4 hours)**

Architecture-client, server, Uniform Resource Locator(URL), Domain Name Service(DNS), Address resolution, Name resolution, Web documents-static document, dynamic document and active documents, Cookies, Virtual hosting, Browser, Browser architecture, HTTP request and response.

Module III: Markup Language **(4 hours)**

Markup language, SGML, HTML, HTML tags and attributes, Cascading Style Sheet (CSS).

Module IV: Web Programming **(3 hours)**

Scripting language, Client Side Scripting Language and Server Side Scripting Language, Writing Java Script.

4. Laboratory Sessions (28 hours)

The practical component of this course is of one credit which amounts to 2 hours of Laboratory classes per week.

- Introduction to HTTP, HTTPS, Web Browsers, URIs.
- Introduction to basic HTML, Class exercise on HTML.
- Introduction to CSS, Internal, Inline and External.
- Client-side scripting language.

5. Referential Sources

Useful Books and Papers

- Tanenbaum, "Computer Network". Pearson India.
- Powell Thomas A., "HTML & CSS The Complete Reference". McGraw Hill.

Useful Web Sources

- https://nptel.ac.in/courses/106/105/106105084/ : NP-TEL content.

Introduction to Latex and R Programming
MCAS2201

1. About the Course

This is a *Skill Enhancement Course* and is aimed to provide a practical introduction to the R programming language and to LaTeX.

2. Course Description

- Target Audience: 2nd semester student of MCA programme
- Total Credit (L + T + P): 2 (0+ 0 + 2)
- Total Credit Hours: 4

- Course Period: One semester (14 Weeks)
    - Total Contact Hours: 56 Hours (14 Weeks X 4 Hours)
    - Practical: 14 Hours (14 Weeks X 4 Hour)

2.1 *Prerequisites and Dependencies*

This course is designed with the beginner in mind. While some participants may have experience in other computer programming languages, no prior programming skills are required.

2.2 *Objective*

The course aims to provide students free and open-source technologies that are used for writing all scientific papers and presentations in Mathematics, by a practical introduction to the R programming language and to LaTeX typesetting language.

2.3 *Course Outcomes*

After course completion, following are the learning/course outcomes.

CO1. Understanding the advantages of LaTeX over other more traditional software.
CO2. Able to use basic components of MiKTeX such as package manager, update manager, etc.
CO3. Able to write mathematical documents via LaTeX having formatting commands and floating bodies, type-in mathematical symbols, equations in paragraph.
CO4. Draws graphs and figures in LaTeX
CO5. Students will be able to download and install R and RStudio
CO6. Navigate and optimise the R integrated development environment (IDE) RStudio
CO7. Learn the main R data structures - vector and data frame and to compute basic summary statistics
CO8. Able to produce data visualisations with the ggplot package

3. Course Contents

Module I: Fundamentals of LaTeX (28 hours)

Document structure, formatting commands, creating: tables, mathematical formulas, theorem, propositions etc., symbols, Graphics in Latex, Pstricks package and figure-graphicdrawing, Creation of table of contents, resource and indexing, Creation of directory of resources with Bibtex.

Module II: Introduction to R and Rstudio (28 hours)

Intro to the R Statistical Programming Language and RStudio IDE, Data importation methods, Basic R Data Types, Data processing and manipulation techniques, Data processing and manipulation techniques, Summary statistic functions, Data visualisations using ggplot, Error types and error handling.

5. Referential Sources

Books:

- Stefan Kottwitz, “LaTeX Beginner's Guide: Create Visually Appealing Texts, Articles, and Books for Business and Science Using LaTeX”. Packt Publishing.
- Norman Matloff, “The Art of R Programming: A Tour of Statistical Software Design”. No Starch Press

SWAYAM Link:

https://onlinecourses.swayam2.ac.in/aic20_sp17/preview : LaTeX Typesetting

Ability Enhancement Courses

Linux Administration
Introduction to Python Programming

Linux Administration
MCAA1201

1. About the Course

This is an *Ability Enhancement Course* and introduces the UNIX/Linux operating system, including: task scheduling and management, memory management, input/output processing, internal and external commands, shell configuration, and shell customization. Explores the use of operating system utilities such as text editors, electronic mail, file management, scripting, and C/C++ compilers.

2. Course Description

- Target Audience: 1st semester student of MCA programme
- Total Credit (L + T + P): 2 (1 + 0 + 1)
- Total Credit Hours: 3
- Course Period: One semester (14 Weeks)
    - Total Contact Hours: 42 Hours (14 Weeks X 3 Hours)
    - Lectures: 14 Hours (14 Weeks X 1 Hour)
    - Practical: 28 Hours (14 Weeks X 2 Hours)

2.1 *Prerequisites and Dependencies*

The course does not have any prerequisite.

2.2 *Objective*

The main objective of this course is to provide knowledge on UNIX/Linux operating system.

2.3 *Course Outcomes*

After course completion, following are the learning/course outcomes.

CO1. Students will be able to identify and use UNIX/Linux utilities to create and manage simple file processing operations, organize directory structures with appropriate security, and develop shell scripts to perform more complex tasks

CO2. Effectively use the UNIX/Linux system to accomplish typical personal, office, technical, and software development

CO3. Monitor system performance and network activities

3. Course Contents

Module I: Introduction **(2 hours)**

Basic nature of the operating system functions and components: Process Control, Memory Management, Input/output control, Security, Popular OS's.

Module II: Terminologies (3 hours)

Command line user interface – CUI, Graphical user interface – GUI, Internal and External commands, Utility programs, Pathing, File transfers, Command processor / command interpreter, File protection, File Encryption, Directory structure, root directory, Programming language translators, Server, Daemon(s), Unix file naming conventions: File names, File extensions, Wild carding (globbing), Absolute and relative paths.

Module III: File Types and Editors (2 hours)

Text vs. binary files, Executable vs. non-executable, Character device files, Block device files, Pipes, Sockets, Directories, Text file editing and formatting using: vi, emacs / pico, ed, nl, pr and X window based WYSIWYG text editors: gedit, leafpad

Module IV: File Processing Operations (3 hours)

Create a file, Access a file using the relative pathname, Access a file using the absolute pathname, Erase or delete a file, Copy a file, Move a file, Cut columns of data from a file, Paste / concatenate files, Rename a file, Create a directory, Display the contents of a directory, Display the user initialization files, Change the working directory, Return to the home directory, Remove a directory, Display the file and/or directory information, Change file/directory permissions, Utilities such as sed and awk, Search files, Search for files by attributes, Sort files

Module V: Shell and C/C++ Programming (2 hours)

Shell initialization files, Aliases, Functions, History mechanism(s) sh, ksh, and bash, Shell variables, Script writing, Script debugging, Script usage, Entering C/C++ programs, Finding syntax errors, Compiling, Source files, Object files

Module VI: Client and Server model in Unix (2 hours)

tcp/ip, udp, Electronic mail services, finger, who / w / users, write / chat, telnet, rlogin, ftp, Web browsers, network administration

4. Laboratory Sessions (28 hours)

The practical component of this course is of one credit which amounts to 2 hours of Laboratory classes per week.

- Introduction to Linux environment.
- Introduction to Terminal and commands.
- Introduction to text based editors like: vi, nl, emacs etc and x-window based editors, gedit, leafpad, etc.
- Shell Programming
- C/C++ Programming
- Network administration in Linux environment

5. Referential Sources

Books:

- Wale Soyinka, “Linux Administration: A Beginner's Guide”. McGraw Hill.
- Patrick H. Wood and Stephen G. Kochan, “Unix Shell Programming”. Sams.

Web Sources:

https://nptel.ac.in/courses/117106113 : NP-TEL content on Linux Basics from IIT Madras.

Introduction to Python Programming
MCAA2201

1. About the Course

This is an *Ability Enhancement Course* that focuses on how to practice and culture the art of programming with Python as a language. The course is designed to explore Python's powerful features, making it ideal for writing effective programs.

2. Course Description

- Target Audience: 2nd semester student of MCA programme
- Total Credit (L + T + P): 2 (1 + 0 + 1)
- Total Credit Hours: 3
- Course Period: One semester (14 Weeks)
    - Total Contact Hours: 42 Hours (14 Weeks X 3 Hours)
    - Lectures: 14 Hours (14 Weeks X 1 Hour)
    - Practical: 28 Hours (14 Weeks X 2 Hours)

2.1 *Prerequisites and Dependencies*

The course is intended to those having experience with high level languages from both procedure and object-oriented paradigms. Prior knowledge of a scripting languages (Perl, PHP, UNIX/Linux shells) is helpful but not mandatory.

2.2 *Objective*

This course leads the students from the basics of writing and running Python scripts to more advanced features such as file operations, working with binary data and using the extensive functionality of Python modules. Extra emphasis is placed on features unique to Python, such as tuples, array slices, and output formatting.

2.3 *Learning Outcomes*

After course completion, following are the learning outcomes.

CO1. Able to develop Python Scripts.
CO2. Implementing object-oriented concepts.
CO3. Build and package Python modules for reusability.
CO4. File I/O.

CO5. Implementing database and GUI applications.

3. Course Contents

Module I: Introduction to Python (2 hours)

Introduction to Python and IDLE to develop programs. Working with datatypes and variables, operators and expressions, working with numeric data, working with string data, Python functions, Boolean expressions, selection structure, iteration structure.

Module II: Working with Data (2 hours)

Lists, indexing and slicing, tuples, dictionaries, references and copies.

Module III: Program Structure (3 hours)

Assignment statements, blocks and syntax rules, expression statements, branching, multiway branching, looping, decisions, continue and break, nested loops control flow.

Module IV: Functions (2 hours)

Defining functions, scope rules, global statements, pass by value vs reference, variable length arguments, argument matching, passing arguments.

Module V: Modules (2 hours)

Imports and attributes, creating modules, namespaces, reloading, module packages, handling files - file I/O, file scanners, files and directories, file positioning, renaming and deleting files.

Module VI: Classes and Objects (3 hours)

An introduction to classes and objects, defining a class, working with object composition, encapsulation, inheritance, exception handling.

4. Laboratory Sessions (28 hours)

The practical component of this course is of one credit which amounts to 2 hours of Laboratory classes per week. Following are the components that will be covered in practical sessions.

- Introduction to editor, translator, IDLE.
- Writing Python scripts.
- Functions in Python.
- File I/O in Python.
- Data analysis modules in Python.
- GUI based application development in Python.

5. Referential Sources

Useful Books and Papers

- Brown Martin C., “Python: The Complete Reference”. McGraw Hill.
- Lutz Mark, “Programming Python”. O’Reilly.
- Allen B.Downey,” Think Python: How to Think Like a Computer Scientist”,O’REILLY

Useful Web Sources

- https://nptel.ac.in/courses/106106145/ : NP-TEL content.

SWAYAM Link

- https://swayam.gov.in/nd1_noc19_cs40/preview : SWAYAM course.

Bridge Courses

Fundamentals of Computer
MCAB1301

1. About the Course

This is a *Bridge Course* and is aimed at presenting foundation concepts of a Computer, its peripherals and various components like Registers, Arithmetic & Logic Unit, Control Unit and Memory etc. The course is organized as a series of lectures, hands-on exercises using Laboratory sessions on various free and open-source software.

2. Course Description

- Target Audience:
  - First semester students of MCA programme not having prior knowledge on computers
- Course Period: One semester (14 Weeks)
- Total Credit (L + T + P): 3 (1 + 1+ 1)
- Total Credit Hours: 3
- Total Contact Hours: 56 Hours (14 Weeks X 4 Hours)
  - Lectures: 14 Hours (14 Weeks X 1 Hour)
  - Tutorial: 14 Hours (14 Weeks x 1 Hour)
  - Practical: 28 Hours (14 Weeks X 2 Hours)

2.1 *Prerequisites and Dependencies*

Since this course is a first semester course, there is no particular prerequisite. However, the students are expected to have elementary knowledge of basic mathematics and logic.
In turn, this course provides foundation for various courses such as “Data Structures using Object Oriented Programming in C++” and “Computer Networking”, which are in the later part of the MCA Programme.

2.2 *Objective*

The major objective of this course is to provide students with understandings of how a computer works, what are its various components, different types of peripherals used,

concept of hardware and software, types of memories in a computer. Basic usage of some free and open-source software is also a major part of the course.

2.3 *Learning Outcomes*

After course completion, following are the learning outcomes.

CO1. Understanding foundation concepts of information and information processing in computer systems.
CO2. Understanding of the basic components of a computer: ALU, CU, Memory, etc.
CO3. Understanding various computer terminologies.
CO4. Differentiation among Hardware, Firmware and Software.
CO5. Understanding the approach to solve a problem.
CO6. Making a student capable to write an algorithm to solve a problem.

3. Topics

Module I: Introduction (6 hours)

What is computer, Computer characteristics, Computer evolutions and generations, Types of computers, Basic components of computer --Control unit, ALU, Input/output device and memory.

Module II: CPU and Memory Unit (8 hours)

CPU architecture-components of CPU, instruction set, register set, types of processor, Memory architecture-storage criteria, primary vs secondary storage, main memory, Cache memory, Secondary storage, Magnetic tape, Magnetic disks , Optical disks, Flash drive.

Module III: Input and Output Devices (6 hours)

Input devices-Keyboard, Scanner, Digitizer, Touch screen etc , Output devices-Monitors, Printers, Plotters.

Module IV: Software and Operating System (8 hours)

Software-concepts and needs, Types of software-system and application software, Algorithm, Flowchart, Pseudo code, Programming Language-Machine language, Assembly language and High-level language, Assembler, Compiler and Interpreter.

4. Laboratory Sessions (28 hours)

The practical component of this course is of one credit which amounts to 2 hours of Laboratory class per week.

4.1 *Section 1: Office Package (LibreOffice)*

- Introduction to Windows 10 OS Environment, Command Prompt, Navigation using GUI based File Explorer (This PC), Concept of Desktop, Icons, Folders and Files, Introduction to Command Prompt. Hands-on various CMD Prompt commands, Environment Variables in Windows.
- Introduction to Linux (Lubuntu) Environment, Concept of Terminal and Terminal commands.
- Introduction to Spreadsheet software: LibreOffice Calc – Layout, Formula Bar, Cell Address, Font design and formatting, Arithmetic Operators (+, –, * and /), Aggregate functions (Avg, Sum, Max, Min and Truncate, etc.), Relational Operators (>, <, >=, <=,

=, !=) Introduction to If .. else statement. Nested If .. else statement, Logical Operators (&, | and !).

- Count If, Introduction to 2D and 3D Graphs – Bar, Pie, Line, Vector, XY Labelling etc.
- Absolute and Reference Cell Addressing, Pivot Table.
- Introduction to Documentation software: LibreOffice Writer – Page Layout, Page Orientation, Page Columns, Font and Paragraph design, Introduction to Numbered and Bullet List and Sub-list, Table design.
- Insertion of Images and shapes, Formatting an Image, Page wrap, Alignment, Insertion of text box. Header and Footer, Page Number, Page Break, Template Design.
- Cover Page design and concept of Mail Merge (using LibreOffice Calc).
- Introduction to Presentation software: LibreOffice Impress – Slide Layout, Master Slide Design, Font and Paragraph Design, Inserting Image and graphs.
- Slide and Custom element Animation. Properties of animation (speed, Event of occurrence, Duration, etc.)

4.2 *Section 2: Multimedia (GIMP, Audacity & OpenShot)*

- Introduction to Image editing software: GNU Image Manipulation Program (GIMP). Environment of GIMP. Concept of Layering, Introduction to various Tools.
- Class exercise on GIMP.
- Introduction to Audio editing software: Audacity, Concept of Monophonic and Stereophonic sound, Concept of channels (L-R), Fading, Combining different Audio tracks.
- Class exercise on Audacity.
- Introduction to Video editing software: OpenShot, Intro to its environment, Concept of tracks, Trimming and Fade effects (Audio and Visual).

5. Referential Sources

Useful Books and Papers

- Rajaraman V. “Fundamentals of Computer”. PHI Publishing.
- Sinha P.K. “Foundation of Computing”. PHI Publishing.
- Byron S Gottfried, “Programming With C”. McGraw Hill
- Brian W. Kernighan, Dennis Ritchie, “The C Programming Language”. Pearson Education India