गोंय विद्यापीठ
ताळगांव पठार
गोंय - ४०३ २०६
फोन: +९१-८६६९६०९०४८

Goa University
Taleigao Plateau, Goa - 403 206
Tel : +91-8669609048
Email : registrar@unigoa.ac.in
Website: www.unigoa.ac.in

(Accredited by NAAC)

GU/Acad –PG/BoS -NEP/2023/184/1 Date:04.07.2023

Ref: GU/Acad –PG/BoS -NEP/2022/339/37 dated 20.08.2022

CIRCULAR

In supersession to the above referred Circular, the updated approved Syllabus with revised Course Codes of the **Master of Computer Applications** Programme is enclosed

The Dean/ Vice-Deans of the Goa Business School are requested to take note of the above and bring the contents of the Circular to the notice of all concerned.

Gaude
(Sanket Gaude)
Offg. Assistant Registrar – Academic-PG

To,

1. The Dean, Goa Business School, Goa University.
2. The Vice-Deans, Goa Business School, Goa University.

Copy to:

1. The Chairperson, Board of Studies in Computer Science and Technology (PG).
2. The Programme Director, Computer Science & Technology Discipline, Goa University.
3. The Controller of Examinations, Goa University.
4. The Assistant Registrar, PG Examinations, Goa University.
5. Directorate of Internal Quality Assurance, Goa University for uploading the Syllabus on the University website.

Goa University

MCA PROGRAMME STRUCTURE effective from Academic Year 2022-23

SEMESTER I – Total 20 credits

Discipline Specific Core(DSC) Courses

Course Code	Course Title	Credits
CSA-500	Data Structures & Algorithms	2
CSA-501	Object Oriented Concepts	2
CSA-502	Operating Systems	3
CSA-503	Internet Technologies	3
CSA-504	Data Structures & Algorithms Lab	2
CSA-505	Object Oriented Programming Lab	2
CSA-506	LINUX Lab	2
	Total Credits for DSC	**16**

Discipline Specific Elective(DSE) Courses – any one to be opted

Course Code	Course Title	Credits
CSA-521	Mathematics for Computer Science	4
CSA-522	Discrete Mathematical Structures	4
	Total Credits for DSE	**4**

SEMESTER II – Total 20 credits

Discipline Specific Core(DSC) Courses

Course Code	Course Title	Credits
CSA-507	Web Development	3
CSA-508	Database Management Systems	3
CSA-509	Machine Learning	4
CSA-510	Web Development Lab	2
CSA-511	Database Management Systems Lab	2
CSA-512	Machine Learning Lab	2
	Total Credits for DSC	**16**

Discipline Specific Elective(DSE) Courses – one to be opted from the DSE list given below

Course Code	Course Title	Credits
CSA-523	Cryptography and Network Security	4
CSA-524	Natural Language Processing	4
CSA-525	Network Programming	4
CSA-526	Human Computer Interaction	4
CSA-527	Agile Methodology	4
CSA-528	Modern Development Platforms	4
CSA-529	Ethical Hacking	4
CSA-530	Advanced Unix Programming	4
CSA-531	Theory of Computation	4
	Total Credits for DSE	**4**

SEMESTER III – Total 20 credits		
Research Specific Electives(RSE) – two to be opted from RSE list given below		
	Total Credits for RSE	**8**
Generic Elective(GE) Courses - total 12 credits to be opted		
Course Code	**Course Title**	**Credits**
CSA-621	Corporate Skills	**4**
	Courses from Other Disciplines for total 8 credits	**8**
	Total Credits for GE Courses	**12**

SEMESTER IV – Total 20 credits		
One Research Specific Elective(RSE) to be opted from the RSE list given below in consultation with the Mentor. It can be completed in Semester 3.		
	Total Credits for RSE	**4**
Dissertation Type		
CSA-651	Research Project in Academic or Research Institutes	
OR		
CSA-652	Industry Internship / Software Project Development	16
Total Credits for Dissertation		**16**

Research Specific Electives(RSE) list		
Course Code	**Course Title**	**Credits**
CSA-600	Speech Processing	**4**
CSA-601	Machine Translation	**4**
CSA-602	Educational Technology	**4**
CSA-603	Computer Graphics	**4**
CSA-604	Data Science	**4**
CSA-605	IoT Architecture and Protocols	**4**
CSA-606	Mobile App Development	**4**
CSA-607	Research Methodology	**4**
CSA-608	Deep Learning	**4**
CSA-609	Programming Paradigms	**4**
CSA-610	Software Testing	**4**
CSA-611	Artificial Intelligence	**4**
CSA-612	MLOps	**4**
CSA-613	IoT Application Development	

SEMESTER I
Name of the Programme: MCA
Course Code: CSA-500
Title of Course: Data Structures & Algorithms
Number of Credits: 2 (2L-0T-0P)
Effective from AY: 2022-23

Prerequisites for the course	Programming using any Programming Language	
Objectives	The aim of the course is to emphasize the importance of data structures in implementing efficient algorithms. It provides an exposure to various algorithm design techniques and an introduction to algorithm analysis.	
Content	**Revision of Programming & Data Structures** Problem solving, Data Types: Primitive and User Defined Selection Constructs, Repetition Constructs, Recursion Pointers Algorithm Representation: - Pseudocode and flowcharts Three level Approach Abstract Data Types (ADTs) Basic Linear Data Structures (LinkedList, Stack, Queue)	5 hours
	Algorithm Analysis Analysis of Algorithms Algorithm Complexity: Space and Time Cases of Complexity: Best, Worst and Average Growth of Functions: Asymptotic Notation	3 hours
	Advanced Linear Data Structures Variants of Linked List and its applications (e.g. Polynomial addition, Sparse matrices) Applications of stacks (e.g. Infix-to-Postfix conversion, Evaluating Postfix Expressions, Bracket Matching) Variants of Queue and Applications	4 hours
	Nonlinear Data Structures: Trees: Binary Search Trees, AVL Trees, B-trees & variants. Tree Traversal Algorithms Heaps and its applications (e.g. implementation of Priority Queue) Graph: Adjacency Matrix and Adjacency List Representations Graph Traversal Algorithms: Breadth First Search and Depth First Search	10 hours
	Divide & Conquer Strategy Algorithms based on Divide and Conquer Strategy: Sorting Algorithms (QuickSort, MergeSort) Binary Search	3 hours
	Greedy Algorithms Huffman Coding Algorithm Minimum Cost Spanning Tree (Prim's, Kruskal's) Single Source Shortest Path (Dijkstra's)	2 hours
	Dynamic Programming Coin Change Problem Longest Common Subsequence All-pair shortest Path (floyd-warshall)	3 hours
Pedagogy	● Lectures/Tutorials/Assignments/Quizzes ● Each data structure should be explained along with implementation of its ADT, its applications and complexity	
References/ Readings	1. Horowitz, Ellis, Sartaj Sahni, and Susan Anderson-Freed. "Fundamentals of data structures in C" WH Freeman & Co., Latest Edition. 2. Thomas H. Cormen, Charles E. Leiserson, et al "Introduction to Algorithms",	

	Latest Edition 3. Allen, Weiss Mark. Data structures and algorithm analysis in C. Pearson Education India, Latest Edition. 4. Dasgupta, Papadimitriou, and Vazirani,Algorithms, by McGraw-Hill. 5. Jeri R. Hanly and Eliot B. Koffman “Problem Solving and Program Design in C” Pearson Education, VII Edition, 2012 6. R.G.Dromey "How to Solve it by Computer ", PHI , Latest Edition
Course Outcomes	Upon successful completion of the course, a student will be able to ● Implement common data structures such as lists, stacks, queues, graphs, and binary trees for solving programming problems. ● Identify and use appropriate data structures in the context of a solution to a given problem. ● Be able to analyze the complexity of a given algorithm

(Back to top)

Name of the Programme: MCA
Course code: CSA-501
Title of course: Object Oriented Concepts
Number of credits: 2 (2L-0T-0P)
Effective from AY: 2022-23

Prerequisites for the course	Knowledge of Programming using any Programming Language	
Objectives	Aim of this course is to introduce the learner to the object oriented paradigm.	
Content	**Classes and objects** Programming paradigm; procedural to object oriented Class; attributes & methods; classes as modules & types; uniform type system, wrapper type classes Object; object references; objects instantiation & interaction; constructor & destructor; pass-by-reference & pass-by-value Object copying & cloning; composite objects Static & non-static members Enumeration & Annotations	8 hours
	Object oriented principles Encapsulation Inheritance; types of inheritance; diamond problem Abstraction; virtual methods Polymorphism; overloading and overriding	8 hours
	Object oriented features Interfaces Access modifiers Errors & Exceptions; user-defined exceptions Collections Anonymous & Inner classes Type parametric polymorphism (e.g. Generics in Java & Templates in C++)	8 hours
	Advanced features Persistence & Serialization; JSON User packages & custom libraries; reflection Predicates & streams Lambda functions	6 hours
Pedagogy	Hands-on assignments / tutorials / peer-teaching / flip classroom. Concepts can be explained using UML class diagrams.	
References/ Readings	**Main Reading** 1. Timothy Budd, “An Introduction to Object Oriented Programming”, Pearson Education, 3rd Edition 2. Brett D. McLaughlin, Gary Pollice & David West, “Head First Object-Oriented Analysis Design”, O’Reilly 3. Ken Arnold, James Gosling, David Holmes, “The Java Programming Language”, Addison-Wesley Professional 4. Stanley Lippman, “C++ Primer”, Addison Wesley 5. Cay S. Horstmann, “Core Java Volume I—Fundamentals”, Pearson 6. Herbert Schildt, “Java: The Complete Reference”, Oracle Press 7. Joshua Bloch, ”Effective Java”, Addison Wesley 8. Kathy Sierra & Bert Bates, “Head First Java”, O’Reilly 9. Bjarne Stroustroup, “The C++ Programming Language”, Addison Wesley 10. https://www.tutorialspoint.com/java/index.htm	
Course	1. Learner will appreciate mapping real-world scenarios in the	

Outcomes	object-oriented world 2. Learner will understand object-oriented principles 3. Learner will be able to design object oriented softwares 4. Learner will be able to analyse a given problem and breakdown into logical units and solve via a bottom-up approach	

(Back to top)

Name of the Programme: MCA
Course Code: CSA-502
Title of the Course: Operating System
Number of Credits: 3 (3L-0T-0P)
Effective from AY: 2022-23

Prerequisites for the course	Computer Architecture Basics	
Objectives	This course focuses on the principles and understanding of the functionality of an operating system and evaluates their trade-off in various environments.	
Content	**Introduction and Systems Structures** Computing Environments, Operating-systems Services, System Calls, System Programs, Virtual Machines, monolithic and micro kernel architectures	3 hours
	Process Management Process - Concept and states, Process Creation and Control, Scheduling Criteria, Scheduling Algorithms, MultiLevel Queues, Multiple-processor scheduling, Real time CPU scheduling	5 hours
	Threads Motivation and Challenges, Multithreading Models, Threading Issues, Thread libraries, Thread scheduling	5 hours
	Process Synchronization Cooperating processes and Race Conditions, The critical-section problem, Peterson’s solution, mutex locks, Synchronization Hardware, Semaphores and their Implementation, Classic problems of synchronization	5 hours
	Inter process Communication, Overview of IPC, Examples of IPC Systems, Communication in Client Server Systems.	3 hours
	Deadlocks System Model, Deadlock characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery From Deadlock	5 hours
	Memory Management Hardware Support, Address Binding, Swapping , Contiguous Memory Allocation, Fragmentation, Memory Protection, Paging, Structure of the page table, Segmentation, Example: Intel architecture	5 hours
	Virtual-Memory Management Background, Demand Paging, Copy-on-write, Page Replacement algorithms, Allocation of Frames, Thrashing, Allocating Kernel Memory	5 hours
	File System File Concept, Access Methods, Directory Structure, File-system mounting, File sharing, Protection. Virtual file systems, Implementing File Systems, Directory implementation, Allocation Methods, Free-space Management, Efficiency and performance, Recovery, Log-structured file systems	5 hours
	Secondary-storage Structure Overview of Mass-storage Structure, Disk Structure, Disk Attachment, Disk Scheduling, Disk Management, Swap-Space Management	4 hours
Pedagogy	lectures/ tutorials/assignments/class presentations and debates/peer reviews/self-study.	
References/	**Main Reading**	

Readings	1. Silberschatz ,Galvin and Gagne , Operating systems Principles – 8th edition or Later(Wiley Asia Student Edition) 2. Deitel H.M., “An Introduction to Operating Systems”, Addison Wesley Publishers Company, Latest Edition 3. Milenkovic M., “Operating Systems : Concepts and Design”, McGraw Hill International Edition Computer Science series ; Latest Edition 4. Tanenbaum A. S., Modern Operating Systems”, Prentice Hall of India Pvt. Ltd.,Latest Edition 5. Operating Systems – a modern perspective - Gary Nutt , Addison Wesley, Latest Edition	
Course Outcomes	1. To understand the services provided by and the design of an operating system. 2. To understand the structure and organization of the file system. 3. To understand what a process is and how processes are synchronized and scheduled. 4. To understand different approaches to memory management. 5. Students should be able to understand the implementation and use of system calls for managing processes, memory and the file system. 6. Students should understand the data structures and algorithms used to implement an OS. 7. Evaluate operating system implementations	

(Back to top)

Name of the Programme: MCA
Course Code: CSA-503
Title of the Course: Internet Technologies
Number of Credits: 3 (3L-0T-0P)
Effective from AY: 2022-23

Prerequisites for the course	Programme requisites	
Objectives:	The objective of the course is to introduce the TCP/IP architecture and allied protocols of the Internet by following a top-down approach.	
Content:	**Computer Networks and the Internet:** Networking and Inter-networks, Internetworking devices, Internet: Network edge, and the Network core. TCP/IP protocol stack: Protocol stack, Connection-oriented, connectionless services, Packet switching, circuit switching, Delay, Loss, and Throughput in Packet-Switched Networks.	6 hours
	Application layer: Principles of Application Layer Protocols, the Web and HTTP, MIME, mail access protocols, DNS, Peer to Peer Applications, Video Streaming, and Content Distribution Networks.	8 hours
	Transport layer: Transport-layer services, Multiplexing and demultiplexing, UDP protocol, Principles of reliable data transfer, Connection-oriented transport - TCP protocol, Principles of congestion control, TCP congestion control.	8 hours
	Network layer: Packet switching: virtual circuit & datagram networks, Forwarding and Routing (Network Data and control planes). The Internet Protocol (IP): IPv4 Datagram format, fragmentation, IPv4 Addressing in the Internet, route aggregation, subnetting, CIDR, Network Address Translation, DHCP, ICMP. Control Plane: Routing protocols- shortest path, link state routing algorithm, distance vector routing. Autonomous Systems (AS), Intra-AS Routing in the Internet: OSPF, Internet routing: RIP, OSPF, BGP, Address Resolution Protocol (ARP), and RARP.	12 hours
	Wireless and Mobile Networks: WiFi (802.11 Wireless LAN), Bluetooth, and Cellular Internet Access.	5 Hours
	Security in Computer Networks: Basic cryptography concepts, Secure Socket Layer (SSL), Internet Security Protocol (IPSec), Virtual Private Network (VPN).	6 hours
Pedagogy:	lectures/ tutorials/assignments/self-study/ flipped classroom	
References/ Readings	1. Forouzan, Behrouz A., and Firouz Mosharraf. “Computer networks: a top-down approach”. McGraw-Hill, 2012. 2. Andrew S. Tanenbaum., “Computer Networks”, (5th Edition) Prentice Hall of India. 3. James F. Kurose, Keith W. Ross, “Computer Networking: A Top-Down Approach” Pearson, Sixth Edition 2017.	
Course Outcomes	After completion of this course, students will be able to • Have a good understanding of layered communication architecture (TCP/IP) and knowledge of some of the important networking protocols • Understand the concepts of reliable data transfer and how TCP implements these concepts. • Basic knowledge of routing algorithms. • Basic knowledge of security in computer networks.	

(Back to top)

Name of the Programme: MCA
Course Code: CSA-504
Title of Course: Data Structures & Algorithms Lab
Number of Credits: 2 (0L-0T-2P)
Effective from AY: 2022-23

Prerequisites for the course	Programing Knowledge	
Objectives	To develop skills to design and implement linear and nonlinear data structures and to identify the most appropriate data structure for solving a real world problem.	
Content	**Lab Assignments may be based on the following**	
	Advanced Linear Data Structures Infix-to-Postfix conversion, Evaluating Postfix Expressions, Bracket Matching	8hours
	Non-linear data structures Binary Trees Tree Traversal Algorithms Binary Search Trees Heap Priority Queue using Heap Heap Sort Graph implementation using Adjacency list and matrix Graph Traversal Algorithms	20hours
	Divide & Conquer Strategy MergeSort QuickSort Binary Search Algorithm	12hours
	Greedy Algorithms Huffman Coding Algorithm Prims' and Kruskal's Algorithm Dijkstra's Algorithm	12hours
	Dynamic Programming Coin Change Problem Longest Common Subsequence Floyd-Warshall Algorithm	8hours
	A Mini Project	
Pedagogy	Programming assignments/ discussions/ self-review/ peer-review/ testing of code/ debugging of code/ projects	
References/ Readings	1. Horowitz, Ellis, Sartaj Sahni, and Susan Anderson-Freed. "Fundamentals of data structures in C" WH Freeman & Co., Latest edition. 2. Thomas H. Cormen, Charles E. Leiserson, et al "Introduction to Algorithms", Latest Edition 3. Allen, Weiss Mark. "Data structures and algorithm analysis in C." Pearson Education India, Latest Edition. 4. Dasgupta, Papadimitriou, and Vazirani, "Algorithms" McGraw-Hill. 2017	
Course Outcomes	Upon successful completion of the course, a student will be able to ● Implement common data structures such as lists, stacks, queues, graphs, and binary trees for solving programming problems. ● Identify and use appropriate data structures in the context of a solution to a given problem.	

(Back to top)

Name of the Programme: MCA
Course code: CSA-505
Title of course: Object Oriented Programming Lab
Number of credits: 2 (0L-0T-2P)
Effective from AY: 2022-23

Prerequisites for the course	Basic Programming Skills	
Objectives	To impart programming skills using object oriented paradigms.	
Content	**Understanding Object Oriented Programming** Suggested sample (non-exhaustive) assignments using an OO visual programming platform like Greenfoot/Alice:- ● Given a game scenario and conditions, create a game and check/modify the OO code generated (e.g. Racing game, Archery, etc.) Suggested sample (non-exhaustive) assignments using an OO language like Java/C++/C# (No CLI input. All values hardcoded in the main method.):- ● Write a procedural program in the OO language (to familiarize with the syntax) and convert the same to an OO code	12 hours
	Applying Object Oriented Principles Suggested sample (non-exhaustive) assignments using an OO language like Java/C++/C# (No CLI input, all values hardcoded in the main method.):- ● Write source code for OO design of a board game (e.g. Chess, Solitaire, etc.) ● Write source code for OO design of an outdoor game (e.g. Football, Tennis) ● Write source code for OO design of your house and allow navigating in the house.	24 hours
	Leveraging the OO features provided by languages Various lab assignments can be given demonstrating the use of the feature and advanced features in the attached 'Object Oriented Concepts' course.	12 hours
	Mini-Project	12 hours
Pedagogy	Hands-on assignments / pair programming / group project/ git project management.	
References/ Readings	**Main Reading** 1. Timothy Budd, "An Introduction to Object Oriented Programming", Pearson Education, Latest Edition. 2. Brett D. McLaughlin, Gary Pollice & David West, "Head First Object-Oriented Analysis Design", O'Reilly, Latest Edition. 3. Ken Arnold, James Gosling, David Holmes, "The Java Programming Language", Addison-Wesley Professional, Latest Edition 4. Stanley Lippman, "C++ Primer", Addison Wesley, 2012 5. Cay S. Horstmann, "Core Java Volume I—Fundamentals", Pearson, 2018 6. Herbert Schildt, "Java: The Complete Reference", Oracle Press, latest edition 7. Joshua Bloch, ”Effective Java”, Addison Wesley 8. Kathy Sierra & Bert Bates, "Head First Java", O'Reilly, 2012 9. Bjarne Stroustroup, "The C++ Programming Language", Addison Wesley, Latest Edition	

	10. https://www.tutorialspoint.com/java/index.htm	
Course Outcomes	1. Learner will be able to write good object oriented code 2. Learner will understand object-oriented principles 3. Learner will be able to design object oriented softwares	

(Back to top)

Name of the Programme: MCA
Course Code: CSA-506
Title of the Course: LINUX Lab
Number of Credits: 2 (0L-0T-2P)
Effective from AY: 2022-23

Prerequisites for the course	Program Prerequisites	
Objectives:	The objective is to introduce students to the Linux operating system environment and provide knowledge of basic Linux commands and shell scripting and system call API.	
Content:	**LINUX Environment** Linux Installation and disk partitioning. Shell, Linux commands, Internal and External Commands, using the documentation/manual, users in Linux: user id, effective user id, use of commands su, sudo, id Basic commands: echo, who, whoami, date, cal, ls, passwd, history, shutdown. Input and output redirection operators (<,<<, >, >>)	12 hours
	The Linux File System, File and Directory management Structure of LINUX file system. Parent-child relationship. Concept of Home directory, current working directory and referring to home directory. Special Files: . and .. Absolute and relative pathnames. Use of PATH variable, Use of command: mkdir, rmdir, pwd, ls and cd. Use of file management commands: nano, touch, cat, cp, mv and rm. FIND command: Searching for a file using find, Finding List of files and directories. Concept of hard disk partitions, file system, Superblock and Inodes, General structure of Linux inode. use of stat command. Analysing the output of ls -l command. File type and permission. Use of chmod command. File ownership: Changing ownership using chown and chgrp commands. Modification and access times. Default file and directory permissions. Use of umask command. Concept of symbolic links. Hard and soft links. Use of ln command to create hard and soft links. Use of commands du, df, tar, zip, gzip, type, which	12 hours
	Filters: File commands- sort, wc, uniq, comm, cmp, diff, pg, tail, head, less, and more , Cut and Paste command Shells' sequence of interpretation of a command; Connecting commands with pipes Regular expressions: grep & sed command AWK script: Selection criteria and action- The BEGIN and END sections, Splitting a line into fields and using printf. Getline function and reading input from files. Writing output to file and pipes. Awk system variables. Using regular expressions. Relational and Boolean operations. Command line parameters and environment variables. Programming constructs: if, for, while.	16 hours

	Process Management Concept of UNIX process. Role of init in process creation. Process ID and exit status of a process. Displaying process attributes using ps command, Killing processes, foreground and background processes. Use of commands job, fg, bg **Package management:** Installing & removing packages	4 hours
	Shell Script Shell scripts and execution methods. The dot command, Interactive and Non Interactive execution. Use of export command, Aliases and command history. Shell variables, Special variables, Built-in shell parameters. Command line arguments. Escaping and quoting. Difference between single and double quotes. Command substitution, brace and tilde expansion, I/O using read and echo. Escape sequences, 'test' command, arithmetic expressions, operators, Control flow: For, If, While, Case. Shell functions, error handling, debugging.	16 hours
Pedagogy:	Practical/ tutorials/assignments/self-study	
References/ Readings	1. Unix Concepts and Applications – Sumitaba Das, Tata MacGraw Hill. 2. Unix and Shell Programming – Graham Glass and King Ables Pearson Education 3. UNIX man pages	
Course Outcomes	Upon completion of this course, the student will be able to: 1. Run various LINUX commands 2. Write shell script on LINUX OS. 3. Use various advanced LINUX tools such as grep, SED and AWK	

(Back to top)

Name of the Programme: MCA
Course code: CSA-521
Title of course: Mathematics for Computer Science
Number of credits: 4 (4L-0T-0P)
Effective from AY: 2022-23

Prerequisites for the course	Program prerequisites	
Objectives	● To build a strong foundation in maths required for learning computer science/data science subjects. ● To understand fundamental concepts and tools in linear algebra etc with emphasis on their applications to computer science in particular data science/machine learning	
Content	**Mathematical logic:** Statement (Proposition), Logical Connectives, Conditional, Bi-conditional, Converse, Inverse, Contrapositive, Exclusive OR, NAND, NOR, Tautology, Contradiction, Satisfiable, Duality Law, Algebra of propositions.	8 hours
	Functions and Relations: Basics of Set theory, Application of set theory, Relations and their properties, n-ary relations and their applications, representing relations, closures of relations, equivalence relations, partial orderings. Functions, properties of functions, Composition of Functions, Recursive functions.	10 hours
	Graphs: Basic Concepts of Graphs, Computer Representations of Graphs, Isomorphic Graphs, Paths, Cycles and Circuits, Eulerian and Hamiltonian Graphs, Planar Graphs, Graph Coloring, Applications of Graphs. Trees: Trees, Spanning trees, Minimal Spanning Trees, Rooted Trees, Binary Trees, Binary Search Trees.	12 hours
	Linear Algebra Scalars, Vectors, Matrices and Tensors -Multiplying Matrices and Vectors - Identity and Inverse Matrices -Linear Dependence and Span -Norms -Special Kinds of Matrices and Vectors - Eigen decomposition -Singular Value Decomposition -The Moore-Penrose Pseudoinverse - The Trace Operator - The Determinant - Example: Principal Components Analysis. **Numerical Computation** Overflow and Underflow -Poor Conditioning - Gradient-Based Optimization - Constrained Optimization -Example: Linear Least Squares. **Calculus** Functions of a single variable, limit, continuity, differentiability-Mean value theorems, indeterminate forms, L'Hospital's rule-Maxima and minima-Product and chain rule-Taylor's series, infinite series summation/integration concepts-Fundamental and mean value-theorems of integral calculus, evaluation of definite and improper integrals-Beta and gamma functions-Functions of multiple variables, limit, continuity, partial derivatives-Basics of ordinary and partial differential equations.	15 hours
	Probability, Statistics, and Information Theory Why Probability? -Random Variables -Probability Distributions - Marginal Probability - Conditional Probability -The Chain Rule of Conditional Probabilities -Independence and Conditional Independence -Expectation, Variance and Covariance -Common Probability Distributions - Useful Properties of Common Functions - Bayes' Rule - Technical Details of Continuous Variables - Information Theory -Structured Probabilistic Models . **Statistics**	15 hours

	Data summaries and descriptive statistics, central tendency, variance, covariance, correlation-Basic probability: basic idea, expectation, probability calculus, Bayes' theorem, conditional probability-Probability distribution functions: uniform, normal, binomial, chi-square, Student's t-distribution, central limit theorem-Sampling, measurement, error, random number generation-Hypothesis testing, A/B testing, confidence intervals, p-values, ANOVA, t-test-Linear regression, regularization	
Pedagogy	Problem-solving approach and carrying out small project work using MatLab tools	
References/ Readings	1. Kenneth H. Rosen, Discrete Mathematics and Its Applications, Tata McGraw-Hill Pub. Co. Ltd. (latest edition) 2. Sheldon M. Ross, "A First Course in Probability", Pearson Prentice Hall, latest edition. 3. Andy Field, Jeremy Miles, Zoë Field, "Discovering Statistics Using R", SAGE, latest edition 4. Omi M Inouye, "Introductory Calculus For Infants", latest edition 5. Robert S. Witte, John S. Witte, "Statistics", Wiley, latest edition. 6. Gilbert Strang, "Introduction to Linear Algebra", Wellesley-Cambridge Press, Fifth Edition (2016).	
Course Outcomes	Students will be able to: 1. Apply mathematics concepts in the modelling and design of computational problems 2. Gain a deeper understanding of subjects like machine learning/deep learning and other computer science subjects.	

(Back to top)

Name of the Programme: MCA
Course Code: CSA-522
Title of the Course: Discrete Mathematical Structures
Number of Credits: 4 (4L-0T-0P)
Effective from AY: 2022-23

Prerequisites for the course	Programme requisites	
Objectives:	The objective of the course is to introduce concepts of mathematical induction, relations, graph theory and boolean functions.	
Content:	Logic, Propositional equivalences, predicates and quantifiers, nested quantifiers, methods of proof, functions.	6 hours
	Mathematical induction, recursive definitions and structural induction, recursive algorithms, programme correctness, Pigeonhole principle, permutations and combinations.	6 hours
	Recurrence relations, solving recurrence relations, divide and conquer algorithms and recurrence relations, generating functions, inclusion and exclusion, applications of inclusion and exclusion.	12 hours
	Relations and their properties, n-ary relations and their applications, representing relations, closures of relations, equivalence relations, partial orderings.	12 hours
	Introduction to graphs, graph terminology, representing graphs and graph isomorphism, connectivity, Euler and Hamiltonian paths, shortest path problems, planar graphs.	12 hours
	Introduction to trees, applications of trees, tree traversal, spanning trees, minimum spanning trees.	6 hours
	Boolean functions, representing Boolean functions, logic gates, minimization of circuits.	6 hours
Pedagogy:	lectures/ tutorials/assignments/self-study/ flipped classroom	
References/Readings	1. Kenneth H. Rosen, Discrete Mathematics and Its Applications, Tata McGraw-Hill Pub. Co. Ltd. 2. Edgar G. Goodaire and Michael M. Parmenter, Discrete Mathematics with Graph Theory, PHI Learning Pvt. Ltd.	
Course Outcomes	After completion of this course, students will be able to ● Have a good understanding of mathematical induction. ● Understand the concepts of Recurrence relation. ● Inherits fundamental knowledge graph theory. ● Acquire Basic knowledge of boolean functions.	

(Back to top)

SEMESTER II
Name of the Programme: MCA
Course code: CSA-507
Title of course: Web Development
Number of credits: 2 (2L-0T-0P)
Effective from AY: 2022-23

Prerequisites for the course	Knowledge of HTML and basic of CSS; Internet Technologies & required protocols; object oriented programming	
Objectives	This course will introduce the learner to the different website development technologies	
Content	**Introduction** ● Evolution of internet & World Wide Web ● Client-Server Architecture ● Revisit HTML & CSS	1 hour
	Enhancing HTML & CSS ● HTML 5 ● CSS3	2 hours
	Front-end Design ● Good Design Rubrics ● Separation of concerns for HTML & CSS; structure vs visual representation ● HTML DOM ● CSS Box Model, pseudo -classes & -elements, CSS animation ● Adaptive & responsive design, viewport & media queries, mobile-first design ● Introduction to a design library and/or & framework (e.g. Bootstrap)	4 hours
	Client-side Scripting ● Dynamic web pages ● JavaScript, programming features, javascript events & functions ● Manipulating DOM ● Beyond ECMA 4 ● Introduction to a Javascript library and framework (e.g. JQuery, ReactJS)	8 hours
	HTTP & Middle-ware ● HTTP, Request & Response, methods & error code, headers, URL encoding & decoding ● XML, data & XPath ● JSON	3 hours
	Server-side Programming ● Server instance ● Request handling & response creation ● HTML forms & file uploads ● Session management & application data ● Database connectivity ● Introduction to a Server-side library and/or template engine and/or framework (e.g. PHP - Laravel; JSP - Spring)	6 hours
	Advanced Web Development ● Model-View-Controller (MVC) & Model-View-ViewModel and others ● Web service architecture and micro-services ● REST calls, Asynchronous JavaScript and XML (AJAX) ● Independent client-server web development	6 hours

	● Difference between Server-side vs client-side rendering ● Introduction to Web stacks, JAM stack & full stack development	
Pedagogy	Hands-on assignments / tutorials / peer-teaching / flip classroom/ presentations	
References/ Readings	1. Robert W. Sebesta, “Programming the World Wide Web”, Pearson Education 2. https://www.w3schools.com/ 3. Steven Holzner, “HTML 5 Black Book” 4. https://www.tutorialspoint.com/ 5. Frank W. Zammetti, “Modern Full-Stack Development”, Apress 6. Nader Dabit, “Full Stack Serverless”, O’Reilly	
Course Outcomes	1. Learner will be able to make decision on what web technology to use and for what purpose 2. Learner will have fair idea on the popular technologies used in website development 3. Learner will appreciate the architecture of web applications and the design decisions	

(Back to top)

Name of the Programme: MCA
Course Code: CSA-508
Title of Course: Database Management Systems
Number of Credits: 2 (2L-0T-0P)
Effective from AY: 2022-23

Prerequisites for the course	A High-Level Programming Language, Data Structures and Algorithms(CS101), Operating Systems(CS103).	
Objectives	This course will enable the learner to understand the different issues involved in the design and implementation of a database system and provide both theoretical knowledge and practical skills required in the creation and use of a Relational DataBase Management System.	
Content	**Basic concepts:** Database & Database Users, Characteristics of the Database Approach, Database Systems, Concepts & Architecture Data Models(RDBMS, Legacy systems, Object Oriented, NoSQL), Schemes & Instances DBMS Architecture of Data Independence, Database languages & Interfaces	3 hours
	Data Modelling using the Entity – Relationship approach	4 hours
	Relational Model, Languages & Systems Relational Data Model & Relational Algebra Relational Model Concepts Relational Model Constraints, Relational Algebra/Relational Calculus	5 hours
	SQL-A Relational Database Language Data SQL - DDL, DML. Views & Queries in SQL. Specifying Constraints & Indexes in SQL. Nested Subqueries, correlated Subqueries	2 hours
	Advanced SQL Embedded SQL, Dynamic SQL, Triggers and Stored Procedures.	2 hours
	Relational Database Design Function Dependencies & Normalization for Relational Database Functional Dependencies Normal forms based on primary keys (1NF, 2NF, 3NF, BCNF) Covers of Functional Dependencies, Canonical covers. Lossless join and Dependency preserving decomposition algorithms.	5 hours
	Transactions and Recovery Techniques Concept of a transaction, Recovery concepts, Recovery Techniques.	4 hours
	Concurrency Control Serializability, Locking Techniques, Time stamp ordering Granularity of Data items	5 hours
Pedagogy	Hands-on assignments / tutorials / peer-teaching / troubleshooting	
References/ Readings	**Main Reading** 1. Korth, Silberchartz, “ Database System Concepts” McGrawhill Publication. 2. Elmasri and Navathe, “ Fundamentals of Database Systems”, Addison Wesley, New Delhi. 3. Database Management Systems –R. Ramakrishnan, J.Gehrke – T.McGraw Hill 4. Desai B., “ An Introduction to Database Concepts”, Galgotia Publications, New Delhi. 5. 2. Rob,Coronel, “Database Systems (Design, Implementation and Management)” 6. Date C. J. , “ An Introduction to Database Systems”, Publication House, New Delhi.	
Course	1. Understand and evaluate the role of a DBMS in information	

Outcomes	Technology applications in Organizations. 2. Recognise and use logical design methods and tools required in the design of DB applications. 3. Understand the relational database design principles. 4. Implement a database Solution to an IT Platform. 5. Understand the basics of SQL and construct queries using SQL. 6. Develop sophisticated queries to extract information from databases. 7. Use embedded SQL queries in a Host Level Language. Understand how the DBMS manages and recovers from concurrent and multiple transactions.	

(Back to top)

Name of the Programme: MCA
Course Code: CSA-509
Title of the Course: Machine Learning
Number of Credits: 4 (4L+0T-0P)
Effective from AY: 2022-23

Prerequisites for the course	Basic concepts of Linear Algebra, Probability theory	
Objectives:	This course provides students with an in-depth introduction to three main areas of Machine Learning: supervised and unsupervised and reinforcement learning. This course will cover some of the main models and algorithms for regression, classification, clustering and Markov decision processes. Topics will include linear and logistic regression, regularisation, SVMs and kernel methods, ANNs, clustering, and dimensionality reduction ,sequential learning Like HMM and reinforcement learning.	
Content:	1. Introduction:- well posed learning problem – designing a learning system-perspectives and issues in machine learning.	4 hours
	2. Concept learning – concept learning task –notation –inductive learning hypothesis-concept learning as search- version space and candidate elimination algorithm-decision tree –random forest.	6 hours
	3. Linear regression - logistic regression-Support vector machine kernel- Model selection and feature selection-Ensemble methods: Bagging, boosting. Evaluating and debugging learning algorithms.	7 hours
	4. Continuous Latent Variables-Revision of Principal Component Analysis -Maximum variance formulation - Minimum-error formulation - Applications of PCA - PCA for high-dimensional data.	7 hours
	5. Neural Networks -Feed-forward Network Functions –perceptron - Weight-space symmetries -Network Training - Parameter optimization -Local quadratic approximation - Use of gradient information - Gradient descent optimization - Error Backpropagation - Evaluation of error-function derivatives - A simple example - Efficiency of backpropagation .	10 hours
	6. Probabilistic model – The normal distribution and its geometric interpretation-probabilistic models for categorical data -using naïve Bayes model for classification, training a naïve Bayes model - discriminative learning by optimizing conditional likelihood - probability models with hidden variables: Expectation-Maximization, Gaussian mixture model	9 hours
	7. Distance-based models – neighbour and exemplers -nearest-neighbour classification -distance based clustering -K means algorithm, clustering around medoids , silhouetees-hierarchical clustering -from kernels to distances	5 hours
	8. Sequential Data - Markov Models - Hidden Markov Models - Maximum likelihood for the HMM -The forward-backward algorithm - The sum-product algorithm for the HMM -Scaling factors - The Viterbi algorithm.	7 hours
	9. Reinforcement learning – Introduction- learning task-Q learing-non deterministic rewards and actions-temporal difference learning.	5 hours
Pedagogy:	Lectures/ tutorials/assignments/self-study	
References/Readings	Main Reading :- 1.Introduction to Statistical Learning, Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, Springer, 2013. 2. EthemAlpaydin, Introduction to Machine Learning, MIT Press. 3. Richard O. Duda, Peter E. Hart, David G. Stork Pattern Classification,.	

	4. Peter Flach , Machine Learning , Cambridge 5.Christopher M. Bishop,Pattern recognition and machine Learning, springer. 6.Deep Learning, Ian Good fellow, MIT press 7.Tom Michele, Machine Learning, McGraw-Hill.	
Course Outcomes	By the end of the course , students should: ● Develop an appreciation for what is involved in learning from data. ● Understand a wide variety of learning algorithms. ● Understand how to apply a variety of learning algorithms to data. ● Understand how to perform evaluation of learning algorithms and model selection. ● Equips them with a general understanding of deep learning.	

(Back to top)

Name of the Programme: MCA
Course code: CSA-510
Title of course: Web Development Lab
Number of credits: 2 (0L-0T-2P)
Effective from AY: 2022-23

Prerequisites for the course	Hands-on experience working with HTML and basic of CSS; Internet Technologies; object oriented programming	
Objectives	This course will focus on the practical use and aspects of the different website development technologies	
Content	**Web Design Assignments** Suggested Sample (non-exhaustive) Assignments:- ● Create a website on a topic given by the instructor. Evaluating the website with rubrics for good web design. ● Build a website using HTML & CSS by looking at a screenshot/picture of a website component given by the instructor. ● Websites built with tables, forms, images, iframes, etc. ● A website for each of design strategies (fixed, adaptive, responsive, fluid, mobile-first, etc.). ● Assignments using css pseudo-classes & -elements; grid & flex design; understanding the CSS box model & working with the browser developer tools; CSS transformations, transitions & animations ● Assignment to create a website built with Bootstrap based on a topic given by the instructor.	15P
	Client-side Scripting Assignments Suggested Sample (non-exhaustive) Assignments:- ● An assignment for understanding the programming aspects of JavaScript and working with the browser developer tools. The use of the newer features of JavaScript (after ECMA 4) is encouraged. ● An assignment working with regular expressions. A search and filter utility can be built. ● Assignments for form data processing and validation and use of HTML5 form elements. A web page with form and validated data could be put in a table. The code could be written using table DOM methods and/or HTML DOM methods and/or XML DOM methods. ● Assignments using various events (mouse, keyboard, etc. events for the form elements, drag-and-drop, window, browser, etc.). ● A web component built using HTML, CSS & JavaScript based on a existing Bootstrap component (e.g. Accordion) ● Assignment with the use of a JavaScript library (JQuery, AngularJS, ReactJS, etc.)	15P
	Developing a Game with HTML, CSS & JavaScript. The game should have at least 500 lines of (HTML+Javascript) code and make use of various mouse/keyboard events.	4P
	Server-side Programming Assignments Suggested Sample (non-exhaustive) Assignments:- ● Assignments to work with HTTP headers for passing data and meta-data, cookies, localStorage ● Assignments to handle data from web forms; handling the request and response payload ● Assignment to manage web sessions	12P

	● Assignment to develop a CRUD functionality by connecting to a database; AJAX calls	
	Full stack Web Developments Develop a CRUD application with MEAN/MERN stack	2P
	Mini-project Ideally done in a group. It should include design and implementation of a web application. Project implementation should mandatorily be built using a templating engine or programming framework (client-side and/or server-side). Project should also use a design framework (e.g. Bootstrap). Conduct and progress of the project could follow industry practices (e.g. git, scrum etc.).	12P
Pedagogy	Hands-on assignments / tutorials / peer-teaching / projects	
References/ Readings	1. Robert W. Sebesta, “Programming the World Wide Web”, Pearson Education 2. https://www.w3schools.com/ 3. Steven Holzner, “HTML 5 Black Book” 4. https://www.tutorialspoint.com/ 5. Frank W. Zammetti, “Modern Full-Stack Development”, Apress 6. Nader Dabit, “Full Stack Serverless”, O’Reilly	
Course Outcomes	1. Learner will be gain experience and be able to create complete websites 2. Learner will be able to make decision on what web technology to use and for what purpose 3. Learner will appreciate the architecture of web applications and the design decisions	

(Back to top)

Name of the Programme: MCA
Course Code: CSA-511
Title of Course: Database Management Systems LAB
Number of Credits: 2 (0L-0T-2P)
Effective from AY: 2022-23

Prerequisites for the course	Hands-on experience in object-oriented programming.	
Objectives	This course aims at enabling learners to develop a skill set to design and implement a realistic application, representative of a typical real-life software system.	
Content	**Installation of DBMS Softwares**	2 hours
	Data Definition Language(DDL) Statements ● Creating a Database. ● Creating a table, with or without constraints. ● Understanding Data types. ● Altering the structure of the table like adding attributes at a later stage, modifying size of attributes or adding constraints to attributes. ● Removing the table created, i.e Drop table in SQL. ● Creating Sequence (Auto increment field)	4 hours
	Query in Data Dictionary ● To view the structure of the table created by the user. ● To view user information. ● To view integrity constraints. ● Altering Session Parameters	2 hours
	Data Manipulation Language(DML) Statements ● Inserting Data into the table. ● Updating Data into the table. ● Deleting Data from the table.	4 hours
	Simple SQL statements ● Displaying all the attributes and tuples from the table. ● Displaying selected attributes/tuples from the table. ● Using Logical and comparison operators. ● String manipulation ● Date Comparisons	6 hours
	Complex SQL Statements ● Using aggregate functions (using Group by and having clauses). ● Sorting Data. ● Creating SQL Aliases and Views. ● Joins and Nested queries. ● Correlated subquery ● Derived tables ● Given a complex table structure, display records from tables.	14 hours
	Transaction Control Language(TCL) statements ● Transactions could be made permanent in memory ● To rollback the transaction.	2 hours
	Embedded SQL statements ● Loops/ if else statements ● Creating Triggers/Procedures/packages ● ArrayList and Cursor. ● PL/SQL Strings ● PL/SQL Object Oriented ● Exceptions	16 hours

	No SQL	4 hours
	Project ● The analysis of project ● Design (ER diagram and normalized tables) and implementation of a real life project of students choice. ● The project report that they submit consists of (i) Feasibility study (ii) ER Diagrams (iii) Tables normalized in an appropriate normal form with integrity and domain constraints noted. (iv) User Interface Design -Form and Report design , including triggers that may need to be written (v) User Manual Peer reviews of ERDs are held in the class.	6 hours (in class)
Pedagogy	Hands-on assignments / tutorials / peer-teaching / troubleshooting	
References/ Readings	1. Korth, Silberchartz, “ Database System Concepts” McGrawhill Publication. 2. Elmasri and Navathe, “ Fundamentals of Database Systems”, Addison Wesley, New Delhi.	
Course Outcomes	1. Design and implement a database schema for a given problem-domain 2: Create and maintain tables using SQL 3: Populate and query a database 4. Use Transaction Control Language 5. Creating and Using User Defined Data Types 6. Writing Triggers & Stored Procedures 7. Prepare reports 8. Application development using PL/SQL & front end tools	

(Back to top)

Name of the Programme: MCA
Course Code: CSA-512
Title of the Course: Machine Learning Lab
Number of Credits: 2 (0L+0T+ 2P)
Effective from AY: 2022-23

Prerequisites for the course	Course: Mathematics for Computer Science and Programming language background.	
Objectives:	The objective is to learn to build the different machine learning models by doing a set of assignments and mini projects.	
Content:	Introduction to python libraries for machine learning - scikit learn, tensor flow, keras, pytorch, pandas, matplotlib, seaborn, numpy and other relevant libraries.	5 hours
	Four branches of machine learning-supervised, unsupervised, self-supervised, reinforcement, Evaluating machine learning models, Data pre-processing, feature engineering and feature learning, overfitting and underfitting - Numerical Programming fundamentals-finding nearest neighbours via Euclidean distance-splitting data sets into training and testing.	10 hours
	Regression, cross validation and regularization-polynomial regression -model selection on a fixed validation set -Polynomial Regression - Model Selection with Cross-Validation-Polynomial Regression with L2 Regularization - Model Selection with Cross-Validation-Comparison of methods on the test set. Evaluating Binary Classifiers and Implementing Logistic Regression-Binary Classifier for movies reviews-classifying newswires-predicting house prices -Computing the Loss for Logistic Regression without Numerical Issues	10 hours
	Neural Networks and Stochastic Gradient Descent-MLPs with L-BFGS: What model size is effective?-MLPs with SGD: What batch size and step size?-Producing your own figure comparing batch size and learning rate.	10 hours
	Trees and Random Forests for Bag of Words-Code Implementation of Decision Tree Regression-Decision Trees for Review Classification - Random Forests for Review Classification -Comparing Trees to Linear Models for Review Classification.	10 hours
	Implementation of CNN, RNN, LSTM, Implementation of Boltzmann machine and Transformers (BERT, GPT3) .Generative deep learning (GAN).	10 hours
	Project discussions -Classifying Images with Feature Transformations-Classifying Sentiment from Text Reviews-Recommendation Systems via Matrix Factorization-Text summarization - language Translation - Sentimental analysis- speech to text translatioXiv, Explore the keras ecosystem.	5 hours
Pedagogy:	Programming in lab and practical exercises	
References/Readings	1. Hands on machine learning with scikit learn by Aurielien 2. Deep learning with python by Francois 3. Text Analytics with Python: A Practitioner's Guide to Natural Language Processing by dipanjan sarkar. 4. keras: the python deep learning API	

	5. https://www.cs.tufts.edu/comp/135/2020f/assignments.html 6. Python library reference	
Course Outcomes	Students will be able – 1. to collect data and preprocess them 2. choose the suitable machine learning model and 3. study its performance and able to carry out mini project	

(Back to top)

Name of the Programme: MCA
Course Code: CSA-523
Title of Course: Cryptography and Network Security
Number of Credits: 4 (4L-0T-0P)
Effective from AY: 2022-23

Prerequisites for the course	Internet Technologies	
Objectives	1. To understand the basics of Cryptography and Network Security. 2. To be able to secure a message over an insecure channel by various means. 3. To learn about how to maintain the Confidentiality, Integrity and Availability of data. 4. To understand various protocols for network security to protect against the threats in the networks.	
Content	**Foundations of Cryptography and Security** Ciphers and Secret Messages, Security Attacks and Services. Classical encryption techniques.	6 hours
	Mathematical Tools for Cryptography Substitutions and Permutations, Modular Arithmetic, Euclid's Algorithm, Finite Fields, Polynomial Arithmetic.	3 hours
	Design Principal of Block Ciphers Theory of Block ciphers, Feistel Cipher network Structures, DES and triple DES, Modes of Operation (ECB, CBC, OFB, CFB), Strength of DES, AES	9 hours
	Pseudo Random Numbers and Stream Ciphers Pseudo random sequences, Linear Congruential generators, Cryptographic generators, Design of stream Ciphers, RC4.	3 hours
	Public Key Cryptography Prime Numbers and testing for primality. Factoring large numbers, Discrete Logarithms.	3 hours
	Asymmetric Algorithms RSA, Diffie-Hellman, ElGamal, Introduction of Ecliptics curve cryptosystems, Key Management, Key exchange algorithms, Public Key Cryptography Standards.	9 hours
	Hashes and Message Digests Message Authentication, MD5, SHA-3, HMAC	6 hours
	Digital Signatures, Certificate and Standards Digital signature standards (DSS and DSA), Public Key Infrastructures, Digital certificates and Basics of PKCS standards.	6 hours
	Authentication Kerberos , X509 Authentication Service	3 hours
	Web Security protocols IP Security, Transport Layer Security(TLS), Wireless Security,	6 hours
	System Security Intrusion detection , Password management, Firewalls management	6 hours
Pedagogy	Lectures/ Hands-on assignment/tutorials/Presentations	
References/ Readings	Main Reading: 1. Stallings William, " Cryptography and Network Security: Principles and Practises", 5th edition, Prentice Hall 2. Kahate Atul, "Cryptography and Network Security" Tata McGraw-Hill.	
Course Outcomes	1. Provide security of the data over the network. 2. Implement various networking security protocols. 3. Protect any network from the threats in the world.	

Name of the Programme: MCA
Course Code: CSA 524
Title of Course: Natural Language Processing
Number of Credits: 4 (2L-2T-0P)
Effective from AY: 2022-23

Prerequisites for the course	Fundamentals of Artificial Intelligence; Mathematical Foundations for Artificial Intelligence. Machine Learning and Programming background. Introduction to NLP (Theory), Mathematical foundations for AI.	
Objectives	This course will focus on understanding the essentials of Natural Language Processing (NLP), areas in NLP, algorithms, and NLP tasks. Students who complete this course will gain a foundational understanding in natural language processing methods and strategies. They will also learn how to evaluate the strengths and weaknesses of various NLP technologies and frameworks as they gain practical experience in the NLP toolkits available.	
Content	Part I: Foundations of Natural Language Processing Introduction ● Natural Language Processing - Problems and perspectives ● Introduction/Recall to/of probability calculus ○ N-grams and Language Models ○ Markov Models ● Introduction to Machine Learning and Deep Learning ● Recurrent Neural Network Language Models ● The evaluation of NLP applications Corpora ● Corpora and their construction: representativeness ● Concordances, collocations and measures of words association ● Methods for Text Retrieval ● Regular expressions	8 hours
	Part II: Natural Language Processing ● Computational Phonetics and Speech Processing ○ Speech samples: properties and acoustic measures ○ Analysis in the frequency domain, Spectrograms ○ Applications in the acoustic-phonetic field. ○ Speech recognition with HMM and Deep Neural Networks ● Tokenisation and Sentence splitting ● Computational Morphology ○ Morphological operations ○ Static lexica, Two-level morphology ● Computational Syntax ○ Part-of-speech tagging ○ Grammars for natural language ○ Natural language Parsing ○ Supplementary worksheet: formal grammars for NL ■ Formal languages and Natural languages. Natural language complexity ■ Phrase structure grammars, Dependency Grammars ■ Treebanks ■ Modern formalisms for parsing natural languages ● Computational Semantics	16 hours

	○ Lexical semantics: WordNet and FrameNet ○ Word Sense Disambiguation ○ Distributional Semantics & Word-Space models ○ Logical approaches to sentence semantics	
	Part III: Applications and Case studies: ● Solving Downstream Tasks: Document classification, Sentiment Analysis, Named Entity Recognition, Semantic Textual Similarity ● Prompting Pre-Trained Language Models ● Network Embedding	6 hours
	Sample list of Assignments to be carried out during the Tutorial Slots - Assignment -1 -Import nltk and download the ‘stopwords’ and ‘punkt’ packages. Assignment-2 -Import spacy and load the language model. Assignment -3 -How to tokenize a given text? Assignment-4 -How to get the sentences of a text document? Assignment- 5-How to tokenize a text using the \`transformers\` package? Assignment -6 - How to tokenize text with stopwords as delimiters? Assignment- 7- How to remove stop words in a text?	30 hours
	Assignment -8- How to add custom stop words in spaCy? Assignment- 9 -How to remove punctuations? Assignment-10 - How to perform stemming? Assignment -11 -How to lemmatize a given text? Assignment-12 -How to extract usernames from emails? Assignment -13-How to find the most common words in the text excluding stopwords Assignment -14- How to do spell correction in a given text? Assignment -15- How to tokenize tweets? Assignment -16- How to extract all the nouns in a text? Assignment -17- How to extract all the pronouns in a text? Assignment - 18 - How to find similarity between two words? Assignment -19- How to find similarity between two documents? Assignment -20 -How to find the cosine similarity of two documents?	
Pedagogy	Hands-on assignments/tutorials / peer-teaching / pair programming/presentations / mini-project. Lectures / Practical / tutorials / assignments / self-study / mini-project	
References/ Readings	1. Allen, James, Natural Language Understanding, Second Edition, Benjamin/Cumming, 1995. 2. Charniack, Eugene, Statistical Language Learning, MIT Press, 1993. 3. Jurafsky, Dan and Martin, James, Speech and Language Processing, Second Edition, Prentice Hall, 2008. 4. Manning, Christopher and Heinrich, Schutze, Foundations of Statistical 5. Natural Language Processing, MIT Press, 1999. 6. Tamburini, F.. Neural Models for the Automatic Processing of Italian, Bologna: Pàtron. 2022 7. T. McEnery and A. Wilson. Corpus Linguistics, EUP. 2001 **8.** https://corpora.ficlit.unibo.it/NLP/ 9. https://www.machinelearningplus.com/nlp/nlp-exercises/ 10. Deep Learning by Goodfellow, Bengio, and Courville free online 11. Machine Learning — A Probabilistic Perspective by Kevin Murphy	

	online 12. Natural Language Processing by Jacob Eisenstein free online Speech and Language Processing by Dan Jurafsky and James H. Martin (3rd ed. draft)	
Course Outcomes	1. Learners will learn about the concepts in natural language processing. 2. Learners will have a fair idea of different areas in NLP 3. Learners will appreciate the complexities involved in natural language processing. 4. Through lectures and practical assignments, students will learn the necessary tricks for making their models work on practical problems. 5. They will learn how to contribute towards the development of NLP Resources and Tools.	

(Back to top)

Name of the Programme: MCA
Course Code: CSA-525
Title of Course: Network Programming
Number of Credits: 4 (4L-0T-0P)
Effective from AY: 2022-23

Prerequisites for the course	Linux lab, Internet technology, Operating Systems	
Objectives	To introduce the basic concept of network programming in UNIX and Windows OS environments.	
Content	Basic UNIX programming: Overview of process, signal handling and related system calls. Systems calls related to process, user and signal Management. File descriptors and inheritance. Named and unnamed pipes and related system calls.	6 hours
	Elementary Socket Programming: Berkley Sockets Overview, Introduction to sockets, Socket addresses, Basic Socket system calls, Error handling. Concept of Reserved ports, Elementary TCP and UDP socket programming. Socket options. Name and Address Conversion functions. Interface Operations using 'ioctl'.	15 hours
	I/O Operations: Synchronous vs. Asynchronous I/O. I/O Multiplexing using 'select' and 'pselect'., Sockets and signals, Signal driven I/O. Nonblocking I/O: Non blocked 'accept' and 'connect'. Broadcasting and Multicasting. Sending and Receiving Out of Band data using 'select' and signals. Advance I/O functions.	15 hours
	Daemon processes and Inetd Super Server	4 hours
	Network Programming in the .NET Framework: System.Net classes overview, working with URI, IP addresses, DNS class, Requests and responses, authentication, and permission.	6 hours
	Socket programming in .NET Working with sockets in .NET, Asynchronous programming, socket permission, support for IPv6, support for TCP, .NET Remoting, support for UDP, multicast sockets. Network tracing, network information, cache management, security.	8 hours
	Programming applications: Time and date routine, Ping, Trivial file transfer protocol, design of chat application using multicast socket programming.	6 hours
Pedagogy	lectures/ Hands-on assignment/tutorials	
References/ Readings	Main Reading: 1. Steven W.R., Unix Network Programming, Prentice Hall of India. 2. Microsoft Software Developers Network Documentation.	
Course Outcomes	After completing the course, students will be able to: ● Analyze and write socket API based programs ● Design and implement client-server applications using TCP and UDP sockets	

(Back to top)

Name of the Programme: MCA
Course Code: CSA-526
Title of Course: Human Computer Interaction
Number of Credits: 4 (4L-0T-0P)
Effective from AY: 2022-23

Prerequisites for the course	Program Prerequisites	
Objectives	To build human-centered design skills, so that you have the principles and methods to create excellent interfaces with any technology.	
Content	Introduction: Human-Computer Interaction, The Power of Prototyping, Evaluating Designs, The Birth of HCI	8 hours
	Needfinding: Participant Observation, Interviewing, Additional Needfinding	8 hours
	Rapid Prototyping: Paper Prototyping and Mockups, Video Prototyping, Creating and Comparing Alternatives	10 hours
	Heuristic Evaluation: Heuristic Evaluation — Why and How? Design Heuristics	8 hours
	Direct Manipulation and Representations: Direct Manipulation, Mental Models, Representations Matters, Distributing Cognition	10 hours
	Visual Design and Information Design: Visual Design, Typography, Grids and Alignment, Reading and Navigating	8 hours
	Designing experiments: Designing Studies That You Can Learn From, Assigning Participants To Conditions, InPerson Experiments, Running Web Experiments, Comparing Rates.	8 hours
Pedagogy	Hands-on assignments / tutorials / peer-teaching / pair programming / presentations / mini-project	
References/ Readings	1. Alan Dix, Janet Finlay, Gregory D. Abowd, and Russell Beale, Human-Computer Interaction (3rd Edition), Pearson, 2004. 2. Ben Shneiderman and Catherine Plaisant, Designing the User Interface: Strategies for Effective HumanComputer Interaction (5th Edition), 5th ed., Pearson Addison-Wesley, 2009 3. Donald A. Norman, The Design of Everyday Things, Basic Books, 2002	
Course Outcomes	1. Learners will be introduced to the concepts in Human centered design skill.	

(Back to top)

Name of the Programme: MCA
Course Code: CSA-527
Title of Course: Agile Methodology
Number of Credits: 4 (4L-0T-0P)
Effective from AY: 2022-23

Prerequisites for the course	Programming Knowledge	
Objectives	The objective of the course is to provide students with a theoretical as well as practical understanding of agile software development practices and how small teams can apply them to create high-quality software.	
Content	**Introduction to Agile Software Development:** Understanding how traditional software development works and it's problems; Role of Agile practices in the world of software development & Tools used	5 hours
	Agile Project Planning And Management: Requirement Analysis, Estimation techniques, Iteration planning, Introduction to development practices: Test Driven Development(TDD) & Pair Programming, Introduction to QA Practices: Fail Fast & Automated functional testing, Introduction to Continuous Integration	30 hours
	Coding and testing practices: Practicing TDD and pair programming as alternative to traditional documentation; Configuring Continuous Integration tools; Automated function testing in detail, Source Control	15 hours
	Agile Software development and deployment: Iterative and incremental software development, Automated and scripted deployment strategies, Handling change requests	10 hours
Pedagogy	Lectures/ Hands-on assignment/tutorials	
References/ Readings	1. Agile Software Development with Scrum, Ken Schwaber, Mike Beedle, Prentice Hall 2. Agile Estimating and Planning by Mike Cohn, Prentice Hall PTR 3. Continuous Integration: Improving Software Quality and Reducing Risk, Paul M. Duvall, Steve Matys, Andrew Glover, Addison Wesley 4. Leading Lean Software Development: Results Are not the Point Mary Poppendieck , Tom Poppendieck	
Course Outcomes	Student will be able to understand, appreciate and apply Agile practices for Software development as well as in real life	

(Back to top)

Name of the Programme: MCA
Course Code: CSA-528
Title of Course: Modern Development Platforms
Number of Credits: 4 (4L-0T-0P)
Effective from AY: 2022-23

Prerequisites for the course	Programming(Program Prerequisites), Knowledge of OS (CSC-103), Internet Technologies (CSC-104) and Web Development (CSC-201,CSC-205)	
Objectives	This course will focus on the modern development technologies, tools and platforms prevalent in the software development industry	
Content	**Overview** ● Ever-changing development terrain, Importance of development at scale. Emergence of Cloud Services, Devops	2 hours
	Development at scale ● Introduction to API Query ● Introduction to ELK stack	4 hours
	Cloud Computing ● Overview ● Cloud Models - IaaS, PaaS, SaaS, Public/Private/Hybrid Cloud ● Components - Virtualisation & VMs, File Storage, Server Instances, Content Delivery Network, etc. ● Setting up cloud ● Cloud Services ● Case study of any one cloud (e.g. Amazon AWS/ Google Cloud/ MS Azure)	24 hours
	DevOps ● Overview of DevOps: ○ Introduction to DevOps ○ DevOps Lifecycle ○ DevOps Delivery Pipeline ● Continuous Integration/ Continuous Delivery (CI/CD) ○ Introduction to CI/CD ○ Continuous Delivery v/s Continuous Deployment ○ Case study of any one CI/CD tool(CircleCI/Jenkins, etc). Case study should include architecture, pipeline and plugin management ● Configuration Management ○ Introduction to Configuration Management ○ Case study of any one Configuration Management(e.g. Ansible, Chef, etc). Case study should include Infrastructure as Code, Inventory Management, playbooks/cookbooks ● Containerization ○ Introduction to Containerization ○ Container Lifecycle ○ Case study of any one containerization tool (e.g. Docker, etc) which should include namespaces, commands,CLI, image creation, image registry ● Continuous Monitoring ○ Introduction to continuous monitoring ○ Types: Infrastructure Monitoring, Application Monitoring and Network Monitoring ○ Case study on one continuous monitoring tool(e.g. Nagios, Prometheus, etc)	18 hours
	Mini Project Ideally done in a group. Concepts and tools (or similar) learnt in the course will need to be implemented/incorporated.	12 hours

Pedagogy	Hands-on assignments / tutorials / peer-teaching / pair programming / presentations / mini-project	
References/ Readings	1. Frank W. Zammetti, "Modern Full-Stack Development", Apress 2. Nader Dabit, "Full Stack Serverless", O'Reilly 3. Joakim Verona, "Practical DevOps" 4. https://www.elastic.co/guide/index.html 5. https://docs.aws.amazon.com/ 6. https://cloud.google.com/docs 7. https://docs.microsoft.com/enus/azure/?product=featured 8. https://docs.docker.com	
Course Outcomes	1. Learner will learn about the latest tools and platforms used in the software industry 2. Learner will have fair idea on the popular cloud services used 3. Learner will appreciate the different devops tools and why devops is important	

(Back to top)

Name of the Programme: MCA
Course Code: CSA-529
Title of Course: Ethical Hacking
Number of Credits: 4 (4L-0T-0P)
Effective from AY: 2022-23

Prerequisites for the course	Internet Technologies, Operating System, Database Management, Programming Skills	
Objectives	To introduce the students to ethical hacking tools and practices used to protect systems from the wide-ranging impact of data breaches and cybersecurity incidents.	
Content	**Introduction:** The importance of security, The various phases involved in hacking, An overview of attacks and exploit categories, The legal implications.	2 hours
	Footprinting: Introduced to footprinting, Information gathering methodology, Tools used for the reconnaissance phase, countermeasures.	3 hours
	Scanning: Detecting 'live' systems on target network, Discovering services running/ listening on target systems, port scanning techniques, active and passive fingerprinting, Automated discovery tools.	3 hours
	Enumeration: Identifying valid user accounts or poorly protected resource shares, active connections to systems and directed queries, Null Session, NetBIOS Enumeration, SNMP enumeration, Applications and Banners.	3 hours
	System Hacking: Remote password guessing, Eavesdropping, Denial of Service, Buffer overflows, Privilege escalation, Password cracking, keystroke loggers, sniffers, Remote control and backdoors, Port redirection, Covering tracks, Hiding files	5 hours
	Trojans and Backdoors: Defining Trojans and Backdoors, Understanding the various backdoor genres, Trojan tools, Prevention methods and countermeasures, Anti-Trojan software.	2 hours
	Sniffers: Active and Passive Sniffing, ARP Spoofing and Redirection, DNS and IP Sniffing and Spoofing.	4 hours
	Denial of Service: DOS and Distributed DOS Attacks, Types of denial of service attacks, Tools for running DOS attacks, Tools for running DDOS attacks, Denial of Service Countermeasures	3 hours
	Social Engineering: Common Types of Attacks, Online Social Engineering, Reverse Social Engineering, Policies and Procedures, Employee awareness.	3 hours
	Session Hijacking: Spoofing Vs Hijacking, Types of session hijacking, TCP/IP concepts, Performing Sequence prediction, ACK Storms, Session Hijacking Tools.	4 hours
	Web Server Hacking: Web Servers and Common Vulnerabilities, Apache Web Server Security, IIS Server Security, Attacks against Web Servers, Countermeasures	3 hours
	Web Application Vulnerabilities: Common Web Application Security Vulnerabilities, Penetration Methodologies, Input Manipulation, Authentication And Session Management, Tools and Countermeasure.	5 hours
	Password cracking: HTTP Authentication Basic & Digest, NTLM Authentication, Certificate Based Authentication, Forms Based Authentication, Password Guessing, Password cracking Tools.	3 hours
	SQL injection: Exploiting the weakness of Server Side Scripting, Using SQL Injection techniques to gain access to a system, SQL Injection Scripts, Prevention and Countermeasures	3 hours

	Buffer Overflow: What is a Buffer Overflow, Exploitation, CPU / OS Dependency, Understanding Stacks, Stack Based Buffer Overflow, Defense against Buffer Overflows **Hacking wireless networks:** Introduction to 802.1, WEP, Cracking WEP Keys, WPA, WLAN Scanners, WLAN Sniffers, Securing Wireless Networks. **Viruses:** Types of viruses, virus signatures, Anti-virus software, few examples. **Evading Firewalls, IDS and Honeypots:** Intrusion Detection System, Integrity Verifiers, Intrusions Detection, Anomaly Detection, Signature Recognition, Protocol Stack Verification, Application Protocol Verification, Hacking Through Firewalls, Honey Pots.	4 hours 4 hours 2 hours 4 hours
Pedagogy		
References/ Readings	**Main Reading** 1. “Hacking Exposed”, Osborne/ Mc Graw Hill. 2. “Hacking Exposed: Network Security Secrets and solutions”, Osborne/ Mc Graw Hill. 3. “Hacking Exposed: Linux Security Secrets and Solutions”, Mc Graw Hill. 4. “Hacking Exposed: Windows Security Secrets and Solutions”, Mc Graw Hill. 5. “Hacking Exposed: Web Application Security Secrets and Solutions”, Mc Graw Hill/Osborne.	
Course Outcomes	1. Discover the elements of a four-phase penetration test and how the four phases help successfully identify system vulnerability. 2. Learn about the different tools and techniques that hackers—including ethical hackers—employ.	

(Back to top)

Name of the Programme: MCA
Course Code: CSA-530
Title of Course: Advanced Unix Programming
Number of Credits: 4 (4L-0T-0P)
Effective from AY: 2022-23

Prerequisites for the course	Basic knowledge of Programming in C and Operating systems	
Objectives	● Introduces system administration tasks, including software installation, system configuration, and managing user accounts. ● Introduce the concept of UNIX system programming including process, signals and interprocess communication.	
Content	Introduction: Organization of UNIX interface, Programmer interfaces. System call API , Error handling. UNIX standardization. UNIX implementations. Relationship of standards and implementation. File I/O and Directories : File descriptor and basic file I/O calls. Duplicating file descriptors. File Types, File access permissions, Set-user-id and set-group-id bits. Setting file permissions. Changing file ownership. Soft and hard links. Reading directories. Synchronising file contents. Standard I/O library.	15 hours
	Process : Environment of UNIX process. Command Line arguments. Environment variables. Memory allocation. Process relationship, Process groups, sessions, Controlling Terminal, Process related system calls. Foreground, Background Processes and Job control. Orphaned process groups.	15 hours
	Signals: Signal concept, Reliable and unreliable signals, Signal sets, Signal related system calls. Non local jumps. Job control using signals.	10 hours
	Terminal I/O: Special Input Characters. Canonical and Non canonical modes. Terminal Option flags. Getting and setting terminal attributes. Pseudo terminals. Opening and using pseudo Terminals. Advanced I/O: Nonbloking I/O, Record locking. Stream, I/O multiplexing, Memory mapped I/O, Asynchronous I/O.	10 hours
	Inter-process communication: Pipes, Message queues, Semaphores and shared memory.	10 hours
Pedagogy	lectures/ tutorials/Hands-on assignments/self-study	
References/ Readings	1. Steven W R, Advanced Programming in UNIX Environment, Addison Wesley. 2. Unix man pages and Standard C library (libc) Documentation	
Course Outcomes	After completing the course, students will be able to: ● Manage UNIX users, file systems, and devices using root powers. ● Access UNIX file management and process management functions via system calls. ● Develop complex system-level software in the C programming language	

(Back to top)

Name of the Programme: MCA
Course Code: CSA-531
Title of Course: Theory of Computation
Number of Credits: 4 (4L-0T-0P)
Effective from AY: 2022-23

Prerequisites for the course	Programme Prerequisites	
Objectives	1. To give an overview of the theoretical foundations of computer science from the perspective of formal languages 2. To illustrate finite state machines to solve problems in computing.	
Content	General Concepts of Automata Theory: Alphabets Strings, Languages, Grammars, Applications of Automata Theory.	3 hours
	Finite Automata (FA): Introduction, Deterministic Finite Automata (DFA) - definition and notations, language of a DFA. Nondeterministic Finite Automata (NFA)- Definition, language of an NFA, Equivalence of DFA and NFA, Applications of FA. Finite Automata with Epsilon Transitions, Eliminating Epsilon transitions, Minimization of DFA. Finite automata with output (Moore and Mealy machines) and inter-conversion.	12 hours
	Regular Expressions (RE): Introduction, Identities of RE. Finite Automata and Regular Expressions - conversions, Algebraic Laws for Regular Expressions, applications of RE. Regular grammars: Definition, regular grammars, and FA, Proving languages to be non-regular (Pumping lemma), Properties of Regular Language, applications.	10 hours
	Context-Free Grammar (CFG): Definition, Derivations Using a Grammar-Leftmost and rightmost derivation, Parse tree, Applications, Ambiguity in CFG. Minimization of CFG, CNF, GNF, Pumping Lemma for CFL's.	10 hours
	Pushdown Automata (PDA): Definition, Language of PDA- Acceptance by Final State and Acceptance by Empty stack, Equivalence of CFG and PDA, Deterministic PDA, Chmosky normal form of CFG Turing Machines (TM): Formal definition and behavior, Languages of a TM, TM as accepters, and TM as a computer of integer functions, Types of TMs.	15 hours
	Recursive And Recursively Enumerable Languages (REL): Properties of recursive and recursively enumerable languages, Universal Turing machine, The Halting problem, Undecidable problems about TMs. Context-sensitive language and linear bounded automata (LBA), Chomsky hierarchy, Decidability.	10 hours
Pedagogy	lectures/ tutorials/assignments/self-study	
References/ Readings	1. John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, Introduction to Automata Theory Languages and Computation, Pearson Education, India (latest edition) 2. H.R.Lewis and C.H.Papadimitriou, Elements of the Theory of Computation, PHI, (latest edition) 3. J.Martin, Introduction to Languages and the Theory of Computation, TMH (latest edition)	
Course Outcomes	At the end of the course students will be able to: ● use basic concepts of formal languages of finite automata techniques ● design Finite Automata for different Regular Expressions and Languages ● Construct context-free grammar for various languages	

SEMESTER III & IV
Name of the Programme: MCA
Course Code: CSA-600
Title of the Course: Speech Processing
Number of Credits: 4 (2L-2T-0P)
Effective from AY: 2022-23

Prerequisites for the course	CSA521-Mathematics for Computer Science and CSA-509 Machine Learning	
Objectives:	The objective of the course is to study fundamental concepts of automatic speech recognition.	
Content:	Anatomy & Physiology of Speech Organs, The process of Speech Production, The Acoustic Theory of Speech Production, Digital models for speech signals.	6 hours
	Formants of vowels, spectrogram of vowels, Acoustic analysis of vowels.	6 hours
	Basic pattern recognition approaches, Parametric representation of speech, Evaluating the similarity of speech patterns, Isolated digit Recognition System, Continuous digit Recognition System.	6 hours
	Hidden Markov Model (HMM) for speech recognition, Viterbi algorithm, Training and testing using HMMs, Adapting to variability in speech (DTW), Language models.	6 hours
	Issues in speaker recognition and speech synthesis of different speakers. Text to speech conversion, Speech to text system. End-to-end systems.	6 hours
	Assignments during Tutorial slots -	
	Basic tools ● Installation of speech processing tools eg. Praat audacity etc. ● Spectrogram visualization	5 hours
	Phonetics and speech signals ● Introduction to International phonetic alphabets ● Audio signal processing and cleaning ● Annotation of speech signal	8 hours
	Formant analysis ● Formant analysis of vowels ● Nasalisation of vowels	7 hours
	Advance concepts ● Installation of kaldi for building ASR ● Creation of phonetic dictionary ● Creation of language model ● Building ASR system	10 hours
Pedagogy:	Lab assignments/ research paper reading/ discussion/ tools demonstration/ mini project.	
References/ Readings	1. Digital processing of speech signals - L.R Rabiner and S.W. Schafer. Pearson Education. 2. Speech Communications: Human & Machine - Douglas O'Shaughnessy, 2nd ed., IEEE Press. 3. Fundamentals of Speech Recognition. L.R Rabinar and B.H. Juang.	
Course Outcomes	After completion of this course, students will be able to • Have a good understanding of human speech production system • Understand the basics of pattern recognition approaches. • Have knowledge of the concepts in speech recognition.	

(Back to top)

Name of the Programme: MCA
Course Code: CSA-601
Title of Course: Machine Translation
Number of Credits: 4 (2L-2T-0P)
Effective from AY: 2022-23

Prerequisites for the course	Knowledge of Mathematics for Computer Science and Machine Learning will prove beneficial, A previous course on Artificial Intelligence and Natural Language Processing will help; Exposure to Linguistics is useful, though not mandatory	
Objectives:	The objective of the course is to understand and get an insight into the different approaches used for Machine Translation (MT).	
Content:	Introduction: Data-driven MT, MT Approaches, Language divergence, three major paradigms of MT, MT Evaluation,	8 hours
	Bilingual Word Mappings: Combinatorial Argument, One-to-One Alignment, Heuristic and Iterative bases computation, Mathematics of Alignment, Expectation Maximization, IBM models of Alignment	4 hours
	Phrase-Based Machine Translation (PBMT): Need, Examples, Phrase Table, Mathematics of Phrase-Based SMT, Decoding.	10 hours
	Rule-Based Machine Translation (RBMT): Kinds, UNL, Interlingua and Word Knowledge, UNL conversion, Transfer-based MT.	5 hours
	Example-Based Machine Translation (EBMT): Essential steps of EBMT, Text similarity computation, Translation memory, Statistical Machine Translation	3 hours
	Assignments during Tutorial Slots -	
	Assignment 1: Data-driven MT, MT Approaches, Language divergence, three major paradigms of MT, MT Evaluation,	8 hours
	Assignment 2: Bilingual Word Mappings: Combinatorial Argument, One-to-One Alignment, Heuristic and Iterative bases computation, Mathematics of Alignment, Expectation Maximization, IBM models of Alignment	4 hours
	Assignment 3: Phrase-Based Machine Translation (PBMT): Need, Examples, Phrase Table, Mathematics of Phrase-Based SMT, Decoding.	10 hours
	Assignment 4: Rule-Based Machine Translation (RBMT): Kinds, UNL, Interlingua and Word Knowledge, UNL conversion, Transfer-based MT.	5 hours
	Assignment 5: Example-Based Machine Translation (EBMT): Essential steps of EBMT, Text similarity computation, Translation memory, Statistical Machine Translation	3 hours
Pedagogy:	lectures/ tutorials/assignments/self-learning/ flipped classroom	
References/ Readings	1. Machine Translation by Pushpak Bhattacharyya, Chapman and Hall/CRC, February 2015 2. Machine Translation on Coursera by Prof. Alexander Waibel and Jan Niehues https://www.coursera.org/learn/machinetranslation 3. An Open Source Neural Machine Translation System https://opennmt.net/ 4. Bhashini Project – https://bhashini.gov.in/bhashadaan/en/likho-india	
Course Outcomes	After completion of this course, students will - • Understand the Machine Translation Approaches • Understand the differences between Phrase-Based, Rule-Based, and Example-Based Machine Translation • explain, apply, and assess evaluation methods for machine translation;	

	• describe and critically discuss the architecture of machine translation systems; • build their own translation model using existing tools for machine translation and evaluate and analyse the translation results; • compare different types of machine translation strategies, such as rule-based, statistical, and neural machine translation; • implement components of machine translation systems or components used in evaluation or pre-processing

(Back to top)

Name of the Programme: MCA
Course code: CSA-602
Title of course: Educational Technology
Number of Credits: 4 (2L-2T-0P)
Effective from AY: 2022-23

Prerequisites for the course	Web Technology	
Objectives	Course aims at Software Developers who wish to develop technology solutions for using Educational Technology in classroom and online mode. Course will offer students an overview of the theories and practices involved in Educational Technology Students will present examples showing the use of technology for classroom management, administration, teaching and learning. Students will select and evaluate appropriate software and hardware for application in the classroom Students will demonstrate legal and ethical use of technology in the classroom. Students will apply technology to develop higher-order skills and creativity	
Content	Learning theories. Learning objectives and Bloom's taxonomy; constructivist and situated theories of learning; factors affecting and facilitating learning; learning styles	8 hours
	Technologies for creating new resources. Examples include video, multimedia, animations and simulations, Web 2.0/3.0.	4 hours
	Instructional Design (ID). Basic ID models (eg ADDIE model), ID models for e-learning and blended learning (eg Dick and Carey model), online course development using ID. Digital Storytelling	8 hours
	Technologies for content delivery. Examples include Learning Management Systems (e.g. Moodle) classroom management systems (e.g. Jhoomla), Open Education Resources, intelligent tutoring systems.	5 hours
	Case Studies: MOOC such as EdX/Coursera, Swayam-NPTEL	5 hours
	Assignments during Tutorial Slots…	
	Introduction to various types of Education Technology tools.	2 hours
	Content Authoring Tools: eg Raptivity, Articulate	3 hours
	Assessment Tools: Hot Potato,	2 hours
	Concept Mapping Tools: e.g. CMAP, MindMap, Compendium	2 hours
	Visualization Tools: e.g. R, Highcharts	3 hours
	Analytics Tools: e.g. SPSS, R-language, CAQDAS	3 hours
	Learning Management System: e.g. Moodle, Sakai	4 hours
	Educational Data Mining: e.g. Weka, Rapidminer, KNIME	2 hours
	MOOC: e.g. EdX	4 hours
	Collaboration Tools: e.g. Wiki	1 hour
	Tutoring system development. e.g. CTAT, ASPIRE	1 hour
	Animation tools. E.g. Flash, Gimp, Others: Camstudio for the screencast, image editing, audio editing (audacity), video management, etc	3 hours

Pedagogy	Hands-on assignments / tutorials / peer-teaching /active learning	
References/ Readings	● Foundations of Educational Technology: Integrative Approaches and Interdisciplinary Perspectives (Interdisciplinary Approaches to Educational Technology) by J. Michael Spector, Routledge; 2nd edition ● Websites/tutorials for the tools	
Learning Outcomes	● Create a portfolio-like presentation with samples reflecting ways technology can support classroom management, administration, and teaching. ● Create and evaluate products that critique various software and hardware tools for instructional purposes ● List and describe legal and ethical issues for using technology in the classroom	

(Back to top)

Name of the Programme: MCA
Course code: CSA-603
Title of course: Computer Graphics
Number of credits: 4 (2L-2T-0P)
Effective from AY: 2022-23

Prerequisites for the course	Knowledge of linear algebra, geometry and programming	
Objectives	This course will introduce the learner to various concepts in 3D modeling and computer graphics	
Content	**Fundamentals** history of computer graphics, applications, graphics pipeline, physical and synthetic images, synthetic camera, modelling, animation, rendering, relation to computer vision and image processing, review of basic mathematical objects (Points, Vectors, Matrix methods).	6 hours
	Exploring OpenGL/WebGL architecture, primitives and attributes, simple modelling and rendering of two- and three-dimensional geometric objects, indexed and RGB color models, frame buffer, double buffering, GLUT, interaction, events and call-backs, picking	6 hours
	Geometric Transformations homogeneous coordinates, affine transformations (translation, rotation, scaling, shear), concatenation, matrix stacks and use of model-view matrix in OpenGL/WebGL for these operations	6 hours
	Viewing classical three dimensional viewing, computer viewing, specifying views, parallel and perspective projective transformations	4 hours
	Shading light sources, illumination model, Gouraud and Phong shading for polygons. Rasterization- Line segment and polygon clipping, 3D clipping, scan conversion, polygonal fill	4 hours
	Discrete Techniques texture mapping, compositing, textures in OpenGL; Ray Tracing-Recursive ray tracer, ray-sphere intersection	4 hours
	Sample List of Assignments to be carried out during Tutorial Slots- 1) Explore a 3D programming IDE (e.g. Alice 3D). Understand basic graphic concepts like objects, camera, direction, projection, etc. 2) Using OpenGL/WebGL/Canvas, write a program to create basic 2D/3D geometric shapes. Use RGB colors. 3) Using OpenGL/WebGL/Canvas, write a program to work around with basic shape transformations (translate, rotate, scale, skew, etc.). 4) Using OpenGL/WebGL/Canvas, write a program to animate objects/shapes (e.g. bouncing ball). Try to incorporate basic physics laws. 5) Using OpenGL/WebGL/Canvas, write a program to import object models. 6) Using OpenGL/WebGL/Canvas, write a program to show object collision. 7) Using OpenGL/WebGL/Canvas, write a program to add texture to objects. 8) Using OpenGL/WebGL/Canvas, write a program to add a light source and implement shadows. 9) Using a 3D modeling tool (e.g. Blender), explore creating complex objects like pillars, cars, etc.	20 hours

	Mini-Project Ideally done in a group. The project should include design and development of a graphic simulation. There should be some interactivity involved. Objects in simulations should be in 3D. Objects could be designed in 3D modelling tools like blender. The texture to those objects could be added programmatically in the simulation before rendering. (e.g. simulation of solar system)	10 hours
Pedagogy	Hands-on assignments / tutorials / peer-learning / flip classroom / analysis of research (or white) papers	
References/ Readings	● Edward Angel, Interactive Computer Graphics. A Top-Down Approach Using OpenGL (fifth edition), Pearson Education, 2008 ● Donald Hearn and Pauline Baker, Computer Graphics with OpenGL (third edition), Prentice Hall, 2003 ● F. S. Hill Jr. and S. M. Kelley, Computer Graphics using OpenGL (third edition), Prentice Hall, 2006 ● Peter Shirley and Steve Marschner, Computer Graphics (first edition), A. K. Peters, 2010. ● James D Foley, Andries Van Dam, Steven K Feiner, John F Huges, Computer graphics with OpenGL: pearson education ● Xiang, Plastock, Computer Graphics, 2nd edition, Tata McGraw ● Kelvin Sung, Peter Shirley, Steven Baer, Interactive Computer Graphics, Concepts and Applications, Cengage Learning ● M M Raiker, Computer Graphics using OpenGL, Elsevier	
Course Outcomes	Learner will - 1. understand and apply fundamental concepts within computer graphics 2. compare and evaluate the ideas in some fundamental algorithms for computer graphics 3. apply fundamental principles within interaction programming 4. understand fundamental concepts of information and scientific visualization	

(Back to top)

Name of the Programme: MCA
Course Code: CSA-604
Title of the Course: Data science
Number of Credits: 4 (2L-2T-0P)
Effective from AY: 2022-23

Prerequisites for the course	Statistics and probability theory and python programming. Python programming and Data science theory fundamentals.	
Objectives	To get started with basics of Data Science and learn all aspects of data science in its entirety. Main objectives are as under - ● to understand basic process of data science ● Python and Jupyter notebooks ● An applied understanding of how to manipulate and analyze uncurated datasets ● Basic statistical analysis and basic machine learning methods like linear regression . ● How to effectively visualize results using python APIs or tools.	
Content	**Unit -1:** Basics of Data Science: Introduction; Typology of problems-**Data Science in a big data world:** Benefits and uses of data science and big data-Facets of data-The data science process-The big data ecosystem and data science-**The data science process:** Overview of the data science process- Defining research goals and creating a project charter- Retrieving data-Cleansing, integrating, and transforming data-Exploratory data analysis-Build the models-Presenting findings and building applications on top of them. **Unit -2** Mathematics for Data science ● Importance of linear algebra, statistics and optimization from a data science perspective; Structured thinking for solving data science problems. ● Linear Algebra: Matrices and their properties (determinants, traces, rank, nullity, etc.); Eigenvalues and eigenvectors; Matrix factorizations; Inner products; Distance measures; Projections; Notion of hyperplanes; half-planes. ● Probability, Statistics and Random Processes: Probability theory and axioms; Random variables; Probability distributions and density functions (univariate and multivariate); Expectations and moments; Covariance and correlation; Statistics and sampling distributions; Hypothesis testing of means, proportions, variances and correlations; Confidence (statistical) intervals; Correlation functions; White-noise process. **Unit -3** Introduction to Data Science Methods: Linear regression as an exemplar function approximation problem; Linear classification problems. **Unit -4** Handling large data on a single computer ● The problems you face when handling large data-General techniques for handling large volumes of data-General programming tips for dealing with large data sets-Case study 1: Predicting malicious URLs-**First steps in big data-** Distributing data storage and processing with frameworks **Unit 5:** Join the NoSQL movement-Introduction to NoSQL **Unit 6: The rise of graph databases** ● Introducing connected data and graph databases ● Introducing Neo4j: a graph database **Unit 7: Data visualization to the end user**	4 hours 2 hours 2 hours 2 hours 4 hours 4 hours 4 hours

	● Data visualization options ● Crossfilter, the JavaScript MapReduce library ● Creating an interactive dashboard with dc.js ● Dashboard development tools ● Data science Story telling.	4 hours 4 hours
	Assignments to be discussed during the Tutorial slots -	30 hours
	1. Python libraries – Numpy, Matplotlib, seaborn, pandas. 2. Write program to do Exploratory data analysis using the libraries above.- Data collection(Kaggle, github and Machine learning repository),data cleaning (removing missing values, reformatting data etc. 3. Write program to do univariate analysis using tools like Box plot, histogram etc. 4. Write program to do bivariate analysis using tools like scatter plots, box plots. 5. Demo on business intelligence tools -Business intelligence tools help an organization analyze huge chunks of data; they provide insights with actionable recommendations - Tableau, Qlik,splunk,Trillium,Logi analytics, powerBI 6. Write program to implement PCA. 7. Write program to implement SVD 8. Use tools like tableau/Power BI to do Visualizatiation for large data set and create dashboard 9. Mini Project: With the tools of Jupyter notebooks, numpy, pandas, and Visualization, you're ready to do sophisticated analysis on your own. You'll pick a dataset we've worked with already and perform an analysis for this first project 10. Machine Learning: To take your data analysis skills one step further, write program to do basics of machine learning and how to use sci-kit learn - a powerful library for machine learning. 11. Working with Text and Databases: You'll find yourself often working with text data or data from databases. This week will give you the skills to access that data. For text data, we'll also give you a preview of how to analyze text data using ideas from the field of Natural Language Processing and how to apply those ideas using the Natural Language Processing Toolkit (NLTK) library. 12. Final Project: These weeks let you showcase all your new skills in an end-to-end data analysis project. You'll pick the dataset, do the data munging, ask the research questions, visualize the data, draw conclusions, and present your results.	
Pedagogy	Lectures/ Tutorials/Hands-on assignments/Self-study. Lab assignments/ research paper reading/ discussion/ tools demonstration/ mini project.	
References/ Readings	1. Practical statistics for data science by peter bruce and andrew bruce 2. Naked statistics by charles wheelon 3. Business data science by matt taddy 4. Elements of statistical learning by Trevor Hastie, Robert and jerome 5. Python for data analysis 6. Data science and big data analytics -EMC2 7. Hands-On Data Structures and Algorithms with Python — By Dr. Basant Agarwal. 8. 3. The Art of Data Science — by Roger D. Peng and Elizabeth	

	Matsui. 9. . Automate the Boring Stuff With Python: Practical Programming— by Al Sweigart.	
Course Outcomes	At the end of the course, the students will – 1. Enrich one's knowledge with overall basics of data science 2. appreciate Data Science to be able to get started in the direction. 3. Students should be able to carry out mini Data Science projects using python libraries.	

(Back to top)

Name of the Programme: MCA
Course Code:CSA-605
Title of Course: IoT architecture and protocols
Number of Credits: 4 (3L-1T-0P)
Effective from AY: 2022-23

Prerequisites for the course	Internet Technologies, Computer Organization and architecture, Operating Systems.	
Objectives	To understand the fundamentals of Internet of Things and the protocols and standards designed for IoT	
Content		
	Introduction to IoT: Introduction, IoT ecosystem, Applications, Challenges.	4 hours
	Fundamentals: IoT Devices - Sensors, Actuators, and gateways, Basics of the wireless sensor network.	6 hours
	IoT Architecture & Design: oneM2M, IoTWF, Additional Reference Models, Core functional stack, Data Management and compute stack.	6 hours
	Communicating smart objects: Communication criteria, communication models, IoT access technologies – 3GPP MTC, IEEE 802.11, IEEE 802.15, WirelessHART, ZWave, Bluetooth Low Energy, Zigbee Smart Energy, DASH7	8 hours
	IoT Network Layer: IP as IoT network layer, IPv6, 6LoWPAN, 6TiSCH, RPL, CORPL, CARP	7 hours
	IoT Transport and Application protocols: Transport Layer: TCP, UDP, DCCP, SCTP, TLS, DTLS IoT application transport methods, HTTP, CoAP, XMPP, MQTT, AMQP, DDS	10 hours
	Security in IoT: MAC802.15.4, 6LoWPAN, RPL, Application Layer security.	4 hours
	Tutorial Slots -	
	IoT Application Case Studies: Discuss minimum 3 Applications in detail of IoT	15 hours
Pedagogy	lectures/ tutorials/Hands-on assignments/self-study	
References/ Readings	1. David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Robert Barton, Jerome Henry, “IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things”, CISCO Press, 2017 2. Hersent, Olivier, David Boswarthick, and Omar Elloumi, The internet of things: Key applications and protocols. John Wiley & Sons, 2011. 3. Buyya, Rajkumar, and Amir Vahid Dastjerdi, eds. Internet of Things: Principles and Paradigms. Elsevier, 2016.	
Course Outcomes	After completing the course, students will be able to: ● Understand the concepts of the IoT Architecture Reference model ● Identify the IoT networking components and ● Identify protocols.	

(Back to top)

Name of the Programme: MCA
Course code: CSA-606
Title of course: Mobile App Development
Number of Credits: 4 (2L-2T-0P)
Effective from AY: 2022-23

Prerequisites for the course	Knowledge of OS and networking; and web development basics	
Objectives	On completion of this course, the learner should be able to successfully build, debug and deploy android apps.	
Content	**Android OS, Ecosystem & Basics** ● Mobile Platforms & OSs; Approaches to mobile development; Android OS; Android System Architecture; Android App Lifecycle; Play Store ● Intro; Create Your First Android App; Layouts, Views and Resources; Text and Scrolling Views; Resources to Help You Learn ● Debugging your apps; Testing your app; Support libraries, and Backwards Compatibility.	6 hours
	User Interface & Lifecycle ● Screen Sizes; User Interaction - User Input Controls, Menus; Screen Navigation; RecyclerView ● Delightful User Experience; Drawables, Themes and Styles; Material Design; Providing Resources for adaptive layouts ● Testing the User Interface ● Activities and Intents; The Activity Lifecycle and Managing State; Starting Activities with Implicit Intents	14 hours
	Background Tasks & Notifications ● Background Tasks; AsyncTask and AsyncTaskLoader; Connecting to the Internet; Broadcast Receivers; Services ● Triggering, Scheduling, and Optimizing Background Tasks; Notifications; Alarm Manager; Transferring Data Efficiently.	4 hours
	Data Saving, Retrieving, Loading ● Overview to storing data ● Shared Preferences; App Settings ● SQLite; Firebase ● Sharing Data: Content Resolvers and Content Providers ● Using Loaders to Load and Display Data ● Connecting with API service endpoints.	6 hours
	Suggested Sample List of Assignments:- 1) Build an OO system (like elevators in a building, EVM, etc.). Employ use of design patterns (like Adapter, Singleton, Observer, etc.) 2) Creating a Java/Kotlin project using build tool (e.g. Gradle, Maven) 3) Create a hello world android app using IDE (preferably Android Studio). Try deploying on emulator/mobile. Debug using logcat. 4) Create a calculator app (similar to the app installed in the device used during development) 5) Using intents create a game (like a maze). Explore having raster images & vector graphics in the app. 6) Create a CRUD app. Explore the use of various form elements/widgets and fragments. 7) Create a To-Do app. Explore adding the views/view-groups programmatically (e.g. using inflate, recycler view). Use material design in the UI. 8) Create an app accessing data exposed by another app/ service.	20 hours

	Explore BroadcastReceiver, services, etc. 9) Create an app that will run in background and communicate information through status bar/ push-notifications. 10) Create a CRUD app using data stored locally. Explore ROOM, SQLite 11) Create an app to consume an API and populate the layout with appropriate views. 12) Create an app to contain a webapp.	
	Mini-project Ideally done in a group. It should include design and implementation of an android application. Project implementation should mandatorily use at least 2 mobile-specific functionality (to justify as a mobile app and not web app). The GUI of the app should follow design guidelines (e.g. Material/ Flat Design). Conduct and progress of the project could follow industry practices (e.g. UX mocks, git, scrum, etc.).	10 hours
Pedagogy	Assignments / tutorials / peer-learning / troubleshooting/ case studies	
References/ Readings	● Bill Philips & Brian Hardy, “Android Programming: The Big Nerd Ranch Guide” ● Dawn Griffiths & David Griffiths, “Head First Android Development” ● Ian F. Darwin, “Android Cookbook” ● https://developer.android.com ● https://kotlinlang.org ● https://material.io	
Course Outcomes	1. Learner will understand the android ecosystem, android versions & compatibility across them. 2. Learner will be able to design user interfaces specifically to be run native android devices. 3. Learner will be able to evaluate which type of views & widgets are preferable for various use cases. 4. Learner will be able to build and design navigation flows in an app. 5. Learner will be able to connect the app to Android services or apps already available on the device. 6. Learner will be able to build apps that can store data locally or remotely.	

(Back to top)

Name of the Programme: MCA
Course Code: CSA-607
Title of the Course: Research Methodology
Number of Credits: 4 (4L-0T-0P)
Effective from AY: 2022-23

Prerequisites for the course	Nil	
Objectives:	The objective of the course is to introduce the theoretical as well as practical aspects of Research	
	Foundations of Research: Meaning, Objectives, Motivation, Utility. Concept of theory, empiricism, deductive and inductive theory. Characteristics of scientific method – Understanding the language of research – Concept, Construct, Definition, Variable. Research Process Problem Identification & Formulation – Research Question – Investigation Question – Measurement Issues – Hypothesis – Qualities of a good Hypothesis –Null Hypothesis & Alternative Hypothesis. Hypothesis Testing – Logic & Importance Research Design: Concept and Importance in Research – Features of a good research design – Exploratory Research Design – concept, types and uses, Descriptive Research Designs – concept, types and uses. Experimental Design: Concept of Independent & Dependent variables.	15 hours
	Qualitative and Quantitative Research: Qualitative research – Quantitative research – Concept of measurement, causality, generalization, replication. Merging the two approaches. Measurement: Concept of measurement– what is measured? Problems in measurement in research – Validity and Reliability. Levels of measurement – Nominal, Ordinal, Interval, Ratio. Sampling: Concepts of Statistical Population, Sample, Sampling Frame, Sampling Error, Sample Size, Non-Response. Characteristics of a good sample. Probability Sample – Simple Random Sample, Systematic Sample, Stratified Random Sample & Multi-stage sampling. Determining size of the sample – Practical considerations in sampling and sample size.	15 hours
	Data Analysis: Data Preparation – Univariate analysis (frequency tables, bar charts, pie charts, percentages), Bivariate analysis – Cross tabulations and Chi-square test including testing hypothesis of association. Interpretation of Data and results	15 hours
	Paper Writing – Layout of a Research Paper, Software for paper formatting like LaTeX/MS Office. Journals in Computer Science, Impact factor of Journals, When and where to publish? Ethical issues related to publishing, Plagiarism and Self-Plagiarism. Software for detection of Plagiarism . Use of Encyclopedias, Research Guides, Handbook etc., Academic Databases for Computer Science Discipline. Use of tools / techniques for Research: methods to search required information effectively, Reference Management Software like Zotero/Mendeley	15 hours
Pedagogy:	Lecture/Presentations/Assignments/Case Study/	
References/ Readings	1. Business Research Methods – Donald Cooper & Pamela Schindler, TMGH, 9th edition 2. Business Research Methods – Alan Bryman & Emma Bell, Sixth Edition, Oxford University Press. 3. Research Methodology: Methods and Techniques, C.R.Kothari, Second Revised Edition, New Age International Publishers	

	4. Social Science Research: Principles, Methods, and Practices, Anol Bhattarchajee, University of South Florida, Scholar Commons. https://digitalcommons.usf.edu/cgi/viewcontent.cgi?article=1002&context=oa_textbooks
Course Outcomes	After completion of this course, students will – • Understand how to formulate a research problem • Understand data collection and analysis techniques • Understand all aspects related to publishing research papers

(Back to top)

Name of the Programme: MCA
Course Code: CSA-608
Title of the Course: Deep Learning
Number of Credits: 4(2L+2T+0P)
Effective from AY: 2022-23

Prerequisites for the course	Familiarity with linear algebra, probability theory, machine learning , familiarity with python	
Objectives:	This course is aimed at any one who wishes to explore deep learning from scratch. This course offers a practical hands on exploration of deep learning, avoiding mathematical notation,preferring instead to explain quantitative concepts through programming using python API	
Content:	Introduction :- what is deep learning ?- Artificial Intelligence, machine learning and deep learning -learning representation from data-"the deep " in deep learning -understanding how deep learning works -what deep learning has achieved so far. Revision of Fundamentals of machine learning- probabilistic modeling – early neural networks- kernel methods-decision tree, random forest and gradient boosting machines -back to neural networks- what makes deep learning different-the modern machine learning landscape . Four branches of machine learning -supervised -unsupervised-self-supervised – reinforcement learning – evaluating machine learning models – data processing, feature engineering- overfitting and underfitting -universal workflow of machine learning The mathematical building block of neural networks – a first look at neural networks – data representation for neural networks- the gears of neural networks :Tensor operations- the engine of neural networks : Gradient -based optimization. Neural networks – anatomy of neural networks- building blocks of deep learning -models of layers -loss functions and optimizers-keys to configuring the learning process.-introduction to keras -keras,tensor flow, theoano and CNTK – developing with keras -setting up a deep learning workstation -case studies – classification movie reviews – classification newswires -predicting house prices. Deep Learning for computer vision – Introduction to convnets – training convnets from scratch on small data sets – using pre trained convnet – visualizing what convnets learn Deep learning for text and sequences – working with text -one-hot encoding of words and characters -using word embeddings-understanding recurrent neural networks – A recurrent layer in Keras -understanding LSTM and GRU layers- A concrete LSTM example in Keras. Advanced use of recurrent neural networks- A temperature-forecasting problem – preparing the data – a common-sense, non machine learning baseline-using recurrent drop out to fight overfitting- stacking recurrent layers-using bidirectional RNNs – sequences processing with convnets Generative deep learning – text generation with LSTM- deep Dream – neural style transfer-generative images with variational autoencoders- introduction to generative adversarial networks. **Assignments to be carried out during Tutorial Slot -** ● Assignment 1 - Logistic Regression with a Neural Network mindset	2 hours 3 hours 3 hours 3 hours 3 hours 3 hours 3 hours 5 hours 5 hours 10 * 3 = 30 hours

	● Assignment 2 - Planar data classification with one hidden layer ● Assignment 3 - Building your Deep Neural Network: Step by Step ● Assignment 4 - Deep Neural Network for Image Classification: Application ● Assignment 5 – Initialization and performance of model, Regularization and whether it helps eliminate overfitting, Gradient Checking with model used, Optimization Methods used for every model ● Assignment 6- TensorFlow Tutorial ● Assignment 7 - Convolution model Step by Step demo ● Assignment 8 - Convolution model Application for image classification ● Assignment 9- Keras Tutorial - Autonomous driving application - Car Detection, Face Recognition ● Assignment 10 - Art Generation with Neural Style transfer	
Pedagogy:	Lectures/ tutorials/lab assignments/self-study	
References/ Readings	Main Reading :- 1. Introduction to Statistical Learning, Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, Springer, 2013. 2. EthemAlpaydin, Introduction to Machine Learning, MIT Press. 3. Richard O. Duda, Peter E. Hart, David G. Stork Pattern Classification,. 4. Peter Flach , Machine Learning , Cambridge 5. Christopher M. Bishop,Pattern recognition and machine Learning, springer. 6. Deep Learning, Ian Good fellow, MIT press 7. Tom Michele, Machine Learning, McGraw-Hill.	
Course Outcomes	By the end of the course , students will be able to: • understand a wide variety of deep learning algorithms. • understand how to apply a variety of learning algorithms to data. • understand how to perform evaluation of learning algorithms and model selection. • equip themselves with a general understanding of deep learning.	

(Back to top)

Name of the Programme: MCA
Course code: CSA-609
Title of course: Programming Paradigms
Number of credits: 4 (4L-0T-0P)
Effective from AY: 2022-23

Prerequisites for the course	Knowledge of programming	
Objectives	To learn and understand various programming paradigms.	
Content	**Understanding Programming Paradigm** ● Programming paradigm concept, motivation, types and classification of paradigms. ● Factors with respect to programming languages: Binding times and flexibility; Scoping; First class values; Abstraction; Typing; Storage Allocation & Dynamic Memory	4 hours
	Imperative Programming ● Variables and data types; Operators and expressions; Input/Output operations, Decision constructs; Looping constructs ● Procedural (in Python/C) -- blocks & scope; procedures (functions) ● Object Oriented (in Java/C++) -- classes & objects, object-oriented principles (encapsulation, abstraction, inheritance, polymorphism)	4 hours
	Functional Programming (in Haskell/Clojure/Scala) ● Revision of mathematical Functions' concepts ● Side effects; Pure functions ● Type induction ● Defining functions ● Currying; Function composition ● Recursion ● Lazy evaluation; infinite lists ● List comprehensions ● Higher order functions; Folds	20 hours
	Logic Programming (in Prolog/ECLiPSe Constraint language) ● Revision of mathematical Logic concepts ● Programming "without algorithms" ● Logic programming with facts, rules and goals ● Recursion; Lists ● Constraint logic programming; constraints as relationship between variables; solving puzzles (like sudoku)	12 hours
	Event-driven Programming (in Python/.NET) ● Events ● Main loop & callback ● Scheduler & Event handlers; Triggers ● Exception handling ● Reliable eventing ● Asynchronous triggers	8 hours
	Multi-Paradigms and more ● Language support for multi paradigms; Benefits & issues ● Parallel programming -- Data Parallelism (in OpenMP) and Message Passing (in MPI) ● Reactive programming (in Elm/ReactiveX for Java, JS) ● Meta programming (in Lisp) ● Natural Language Programming (in SciLab/MATLAB)	12 hours

Pedagogy	Hands-on assignments / tutorials / peer-learning / pair programming/ analysis of research (or white) papers	
References/ Readings	● Terrance W. Pratt, Marvin V. Zelkowitz, “Programming Languages - Design & Implementation” ● Robert L. Sebesta, “Concepts of Programming Languages” ● Ravi Sethi, “Programming Languages Concepts & Constructs” ● Bruce J. Mac Lennan, “Principles of Programming Languages: Design, Evaluation, and Implementation” ● Kenneth C. Louden, “Programming Languages: Principles and Practice” ● Allen Tucker, Robert Noonan, “Programming Languages: Principles and Paradigms” ● Graham Hutton, “Programming in Haskell” ● W. Clocksin, “Programming in Prolog” ● Slim Abdennadher, Thom Frühwirth, “Essentials of Constraint Programming” ● Roland Kuhn, Brian Hanafee, Jamie Allen, “Reactive Design Patterns”	
Course Outcomes	1. Learner will be able to distinguish between different programming paradigms 2. Learner will be able to choose an adequate programming paradigm in solving specific software engineering problems 3. Learner will be able to recognize the similar concepts implemented in a different way across different programming languages and paradigms	

(Back to top)

Name of the Programme: MCA
Course code:CSA-610
Title of course: Software Testing
Number of Credits: 4 (2L-2T-0P)
Effective from AY: 2022-23

Prerequisites for the course	Software Engineering, OOT, Web Technology, Agile Methodology	
Objectives	• Inculcate the concepts and skills related to testing and quality assurance • To empower the learner to evaluate and select appropriate testing methods and tools • Develop Test first approach to software development. • Inculcate the concepts and skills related to testing and quality assurance. • Use various tools for testing and test automation • To empower the learner to evaluate and select appropriate testing methods and tools.	
Content	Fundamentals of testing: Test, test case, test case design Levels of testing: Unit, Integration, system, Acceptance Testing Types of testing: White box and black box, various techniques – Cyclomatic complexity, equivalence class partitioning, boundary value analysis Functional and non-functional testing.	8 hours
	Test Driven Development: TDD frameworks and refactoring using Junit, pair programming	8 hours
	Debugging approaches and principles, debugging guidelines	4 hours
	Testing tools and frameworks for Web and App development: Selenium, Jmeter, Jira, Bugzilla, API testing, DB testing,	4 hours
	Continuous Integrations and DevOPs	2 hours
	Quality Assurance: Reviews, walkthroughs, quality frameworks	4 hours
	Tools to be discussed during Tutorial Slots -	**10 * 3 = 30 hours**
	Test management tool: keep track of all the testing activity, fast data analysis, manage manual and automation test cases, various environments, and plan and maintain manual testing	3 hours
	Bug tracking tool: commonly used bug tracking tools such as: Jira, Bugzilla	3 hours
	Automated testing tool: how to change the manual test cases into a test script with the help of some automation tools. commonly used automation testing tools: Selenium	3 hours
	Performance testing tool: test the performance of the software or an application. Performance testing tools such as Apache JMeter, LoadRunner	3 hours
	Cross-browser testing tool: to test application on multiple browsers , perform compatibility testing through various browsers by using cross-browser testing tools such as LambdaTest, Sauce Labs	3 hours
	Integration testing tool: test the interface between modules and find the bugs. Some of the most used integration testing tools : Citrus, FitNesse	3 hours
	Unit testing tool using Junit/NUnit/phpunit and refactoring tools	3 hours

	Mobile/android testing tool to check the usability, functionality, security, and consistency of the application. Use of tools of mobile testing such as Appium	3 hours
	GUI testing tool GUI testing:Navigation validation, verify the check screens, data integrity validation, verification of usability situations, and also check the numeric, date field formats.	3 hours
	Security testing tool authorization, confidentiality, authentication, and availability types of aspect SonarQube ZAP	3 hours
Pedagogy	Classroom/handson instructions, assignments, mini projects. Demo of tools, Classroom/handson instructions, assignments, mini projects	
References/ Readings	1. Agile Java: Crafting Code with Test-Driven Development, Prentice Hall; 1st edition, 2005 2.A Practitioner's Guide to Software Test Design, Lee Copeland, Artech House 3. Refactoring: Improving the Design of Existing Code by Martin Fowler, Pearson, 2009 4. Code Complete- Steve McConnel, Microsoft Press US; 2nd edition, 2004 Websites and online tutorials	
Course Outcomes	At the end of the course, the students will be able to – 1. design test cases 2. apply agile and lean principles in software design 3. configure and use various test automation tools 4. adopt best practices in software testing and quality assurance 5. use testing tools for all aspects of software testing 6. evaluate and select appropriate tools for a software project	

(Back to top)

Name of the Programme: MCA
Course Code: CSA-611
Title of the Course: Artificial Intelligence
Number of Credits: 4 (2L-2T-0P)
Effective from AY: 2022-23

Prerequisites for the course	Strong knowledge of Mathematics; Good command over programming languages; Good Analytical Skills; basic knowledge of Statistics and modelling.	
Objectives:	This course provides students with an in-depth introduction to the five main tribes of Artificial Intelligence-namely Symbolists, Connectionists, Bayesians, Evolutionaries and Analogizers. Symbolists systems include Decision trees, Random decision forests, Production rule systems, inductive programming. Connectionists include Artificial Neural nets, Reinforcement learning, Deep learning Bayesians include Hidden Markov Chains-Graphical Models-Causal inference Evolutionary -biologist - biologically inspired computing Analogizers (psychologists) include k nearest neighbour algorithm. This course is aimed at exploring all facets of AI and obtain in-depth understanding of this facilitating field.	
Content:	**Unit-1** :-Introduction to AI :- The roots of Artificial Intelligence - Five tribe of AI -The symbolist - connectionist -Evolutionaries-The Bayesians-Analogizer	1 hour
	Unit-2:-Symbolic Tribe (Symbolic AI) Problem-solving-Solving Problems by Searching -Search in Complex Environments - Adversarial Search and Games -Constraint Satisfaction Problems . Knowledge, reasoning, and planning Logical Agents - First-Order Logic - Inference in First-Order Logic - Knowledge Representation - Automated Planning .	5 hours
	Unit-3 :-Bayesian Tribe :- Uncertain knowledge and reasoning - Quantifying Uncertainty -Probabilistic Reasoning-Probabilistic Reasoning over Time -Probabilistic Programming -Making Simple Decisions -Making Complex Decisions -Multiagent Decision Making	5 hours
	Unit-4:- Connectionism tribe :- Machine Learning - supervised learning -unsupervised learning-Artificial neural networks-perceptron-MLP-deep neural network -CNN-RNN-LSTM -hop field neural network	5 hours
	Unit-5 :- Evolutionaries tribe:- An Overview of Combinatorial Optimization-An Introduction to Genetic Algorithms-Theoretical Foundations of Genetic Algorithms-Genetic Algorithms in Engineering and Optimization-Genetic Algorithms in Natural Evolution-Simulated Annealing and Tabu Search GALib-Genetic Algorithm Optimization Toolbox (GAOT) under Matlab.	5 hours
	Unit-6 :- Analogizers :- constrained optimization ,Margin and SVM-hard margin and soft margin, non-linearity - kernel- different types of kernels-k nearest neighbors	5 hours
	Unit 7 :- Communicating, perceiving, and acting-Natural Language Processing -Deep Learning for Natural Language Processing - Computer Vision -Robotics Conclusions- Philosophy, Ethics, and Safety of AI - Explainable AI - The Future of AI	4 hours
	Problem Solving during Tutorial Slots…	10 * 3 = 30 hours
	1. Real-world path planning for pedestrians. In the first part,	

	students implement A* over a map that includes roads/paths as well as elevations. In the second part, students collect actual data through walking around the real world, and the cost model is then learned via regression techniques. 2. Solve maze via search -this assignment involves formulating maze-solving as a search problem, image processing (via OpenCV) as a step in maze-solving, as well as guided performance/quality analysis of representational parameters. 3. Within the context of an artificial intelligence course, students are taught to identify ethical issues within technical projects and to engage in moral problem solving with regard to such issues. 4. Neural network for face recognition using tensor flow -build feedforward neural networks for face recognition using TensorFlow. Students then visualize the weights of the neural networks they train. The visualization allows students to understand feedforward one-hidden layer neural networks in terms of template matching, and allows students to explore overfitting. 5. Organic path finding -Students develop a “human-like” pathfinding technique by specializing a generic search algorithm with custom action cost and heuristic cost functions. Students apply classical search algorithms and reflect on example organic paths to achieve “human-like” pathfinding. 6. Implement a genetic algorithm in Python to evolve strategies for Robby the Robot to collect empty soda cans that lie scattered around his rectangular grid world. 7. Compare the performances of a brute-force search and a search employing the Minimum Remaining Values (MRV) heuristic in solving Sudoku puzzles. 8. The students need to understand and extend an existing implementation of the back-propagation algorithm and use it to recognize static hand gestures in images. 9. Students learn about feedforward neural networks and the backpropagation algorithm by implementing a perceptron network for AND and XOR Boolean functions and, given an implementation of a feedforward network, learn digit recognition using the MNIST data set. 10. In this assignment students extend a Tic Tac Toe program to Ultimate Tic Tac Toe and implement a different search strategy than the example code.	
Pedagogy:	Lectures/ tutorials/assignments/self-study.	
References/ Readings	Main Reading :- 1. Master algorithm by pedro domingos 2. Artificial Intelligence -Modern approach -Russel and Norvig- 4th Edition 3. Hands on Machine learning with sci-kit learn and tensorflow- Orellie 4. Deep learning with python by Francois - 5. Elements of statistical learning - Trevor Hastie,Robert and Jerome -springer. 6. Bayesian reasoning and machine learning - David barber 7. Genetic algorithm by David E Goldberg.	

	8. Artificial Intelligence- A Modern Approach (3rd edition) by norvig , russel 9. Artificial Intelligence By Example-2nd edition by Denies Rothman,PACKT 10. Artificial Intelligence Engines: A Tutorial Introduction to the Mathematics of Deep Learning 11. Human Compatible: Artificial Intelligence and the Problem of Control by Stuart Russel References 1. Artificial Intelligence - A guide for thinking humans by Melaine Mitchell. 2. A world without work - by Daniel susskind. 3. Genius Makers -Cade Metz 4. what computer still cannot do by Hubert Dreyfus 5. The alignment problem -Brian Christian 6. Clara and sun by Kazuo Ishiguro 7. Rebooting AI by Gary Marcus and Ernest Davis 8. Four futures -Peter Frase 9. Flake, The Computational Beauty of Nature, MIT Press, 1998. 10. von Neumann, The Computer and the Brain. Yale University Press, 1958 11. https://formtek.com/blog/artificial-intelligence-the-five-tribes-of-ai/	
Course Outcomes	By the end of the course , students will be able to - ● understand a wide variety of AI algorithms. ● learn to apply different tribes in different applications. ● understand how to apply a variety of learning algorithms to data. ● understand how to perform evaluation of learning algorithms and model selection. ● further learn to understand the need to understand Master algorithm - unification of all algorithms to solve complex problems. ● carry out the mini project work with respect to the different paradigms.	

(Back to top)

Name of the Programme: MCA
Course Code: CSA-612
Title of the Course: MLOps
Number of Credits: 4 (2L+2T+0P)
Effective from AY: 2022-23

Prerequisites for the course	Familiarity with linear algebra, probability theory, machine learning , familiarity with python.	
Objectives:	This course is aimed at any one who wishes to explore deep learning from scratch. This course offers a practical hands on exploration of deep learning, avoiding mathematical notation,preferring instead to explain quantitative concepts through programming using python API	
Content:	**Unit 1**. Introduction to MLOps Rise of the Machine Learning Engineer and MLOps-What Is MLOps?-DevOps and MLOps-An MLOps Hierarchy of Needs-Implementing DevOps-Configuring-Continuous Integration with GitHub Actions-DataOps and Data Engineering-Platform Automation-MLOps	3 hours
	Unit 2. MLOps Foundations-Bash and the Linux Command Line-Cloud Shell Development Environments-Bash Shell and Commands-List Files Run CommandsFiles and Navigation-Input/Output-Configuration-Writing a Script-Cloud Computing Foundations and Building Blocks-Getting Started with Cloud Computing- minimalistic python revision-Descriptive Statistics and Normal Distributions-Optimization-Machine Learning Key Concepts-Doing Data Science-Build an MLOps Pipeline from Zero	3 hours
	Unit 3. MLOps for Containers and Edge Devices Containers-Container Runtime-Creating a Container Running a Container-Best Practices-Serving a Trained Model Over HTTP-Edge Devices-Coral Azure Percept-TFHub-Porting Over Non-TPU Models-Containers for Managed ML Systems-Containers in Monetizing MLOps-Build Once, Run Many MLOps Workflow	3 hours
	Unit 4. Continuous Delivery for Machine Learning Models-Packaging for ML Models-Infrastructure as Code for Continuous Delivery of ML Models-Using Cloud Pipelines-Controlled Rollout of Models-Testing Techniques for Model Deployment	3 hours
	Unit 5. AutoML and KaizenML-AutoML-MLOps Industrial Revolution-Kaizen Versus KaizenML-Feature Stores-Apple’s Ecosystem-Apple’s AutoML: Create ML-Apple’s Core ML Tools orGoogle’s AutoML and Edge Computer Vision or Azure’s AutoMLor AWS AutoML-Open Source AutoML Solutions-Ludwig-FLAML-Model Explainability	3 hours
	Unit 6. Monitoring and Logging-Observability for Cloud MLOps-Introduction to Logging-Logging in Python-Modifying Log Levels-Logging Different Applications-Monitoring and Observability-Basics of Model Monitoring-Monitoring Drift with AWS SageMaker-Monitoring Drift with Azure ML	3 hours
	Unit 7. MLOps for AWS-Introduction to AWS-Getting Started with AWS Services-MLOps on AWS-MLOps Cookbook on AWS-CLI Tools-Flask Microservice-AWS Lambda Recipes-AWS Lambda-SAM Local-AWS Lambda-SAM Containerized Deploy-Applying AWS Machine Learning to the Real World	3 hours
	Unit 8. Machine Learning Interoperability-Why Interoperability Is Critical-ONNX: Open Neural Network Exchange-ONNX Model Zoo-Convert PyTorch into ONNX -Convert TensorFlow into ONNX-Deploy ONNX to Azure-Apple Core ML-Edge Integration	3 hours
	Unit 9: Building MLOps Command Line Tools and Microservices-	3 hours

	Python Packaging-The Requirements File-Command Line Tools-Creating a Dataset Linter Modularizing a Command Line Tool-Microservices-Creating a Serverless Function-Authenticating to Cloud Functions-Building a Cloud-Based CLI-Machine Learning CLI Workflows **Unit 10**. Machine Learning Engineering and MLOps Case StudiesUnlikely Benefits of Ignorance in Building Machine Learning Models-MLOps Projects at Sqor Sports Social Network-Mechanical Turk Data Labeling-Influencer Rank-Athlete intelligence (AI product)-The perfect techniques versus the real world-critical challenges in MLops- Ethical and unintended consequences-lack of operational excellences- focus on prediction accuracy vs the big picture	3 hours
	Assignments to be done during Tutorial Slots: 1. Perfect Project Structure – Cookiecutter & readme.so 2. Speed Exploratory Data Analysis to Minutes – Pandas Profiling, SweetViz 3. Track Data Science Projects with CI, CD, CT, CM –Data Version Control (DVC) 4. Explainable AI / XAI – SHAP, LIME, SHAPASH 5. Deploy ML Projects in minutes – Docker, FastAPI 6. End to End Machine Learning – MLflow 7. Building Production Ready ML Pipelines - Model Registry, Feature Store (Feast, ButterFlow) 8. Big Data using Python, instead of PySpark – DASK 9. Build a Chatbot and Deploy it (open-source) 10. FaaS Framework implementation – Apache OpenWhisk, OpenFaas	**10 * 3 = 30 hours**
Pedagogy:	lectures/ tutorials/lab assignments/self-study	
References/ Readings	Main Reading :- Practical MLops – Noah Gift and AlfredoDeza Introduction to MLOps – Noah Gift and AlfredoDeza 1| Machine Learning Engineering By Andriy Burkov 2.ML Ops: Operationalizing Data Science By David Sweenor, Dev Kannabiran, Thomas Hill, Steven Hillion, Dan Rope and Michael O'Connell-O'Reilly 3. Building Machine Learning Pipelines By Hannes Hapke, Catherine Nelson 4. Practical MLOps by Noah Gift, Alfredo Deza. O'Reilly 5. Introducing MLOps By Mark Treveil & Dataiku Team 6.Beginning MLOps with MLFlow: Deploy Models in AWS SageMaker, Google Cloud, and Microsoft Azure By Sridhar Alla, Suman Kalyan Adari, O'Reilly	
Course Outcomes	Students will be able to – 1. handle deployment challenges in ML projects 2. develop technical competence to deploy the ML projects.	

(Back to top)

Name of the Programme: MCA
Course code: CSA-613
Title of course: IoT Application Development
Number of Credits: 4 (2L-2T-0P)
Effective from AY: 2022-23

Prerequisites for the course	Programming skills, basic knowledge of electronics, Basics of networking	
Objectives	The basic objectives are: ● To introduce the concept of the Internet of Things and its applications in various domains ● To explore the different protocols and communication methods used in IoT systems ● To provide a working knowledge of Node-RED, a popular programming tool for developing IoT applications ● To equip students with the skills to design and build IoT systems for a variety of use cases	
Content	Fundamentals of IoT ● Understanding IoT and its applications ● IoT architecture and components ● Introduction to sensors and actuators	8 hours
	IoT protocols and communication ● Wired and wireless communication protocols ● Overview of IoT protocols: MQTT, CoAP, HTTP, WebSocket, etc. ● LoRaWAN and its applications	8 hours
	Cloud Computing for IoT ● Cloud computing fundamentals ● Cloud services for IoT ● Cloud platforms for IoT ● IoT data management and storage on the cloud	8 hours
	IoT Security and Privacy ● IoT security risks and challenges ● IoT security protocols and practices ● IoT privacy concerns and regulations	6 hours
	Assignments to be discussed and carried out during the Tutorial Slots	
	Introduction to Node-RED ● features, architecture, and installation ● Building the flow: understanding nodes, messages, and flows ● Debugging the flows: using the debug node, logging, and error handling	6 hours
	Data acquisition and visualization ● Using sensors and actuators in Node-RED ● Connecting to sensors and devices: using input nodes and protocols (MQTT, HTTP, WebSocket, etc.) ● Data processing and manipulation: using function nodes and JavaScript ● Building dashboards: using the Node-RED Dashboard module for data visualization and control ● Using APIs and cloud services in Node-RED	8 hours
	IoT protocols and communication ● Overview of IoT protocols: MQTT, CoAP, HTTP, WebSocket, etc. ● Setting up an MQTT broker: installation, configuration, and security ● MQTT publishing and subscribing: using MQTT nodes in Node-RED	8 hours

	● Building an MQTT-based IoT system: integrating sensors, actuators, and applications	
	Advanced topics in IoT and Node-RED ● Node-RED extensions and plugins ● Deploying and scaling Node-RED: hosting Node-RED flows on cloud platforms like AWS IoT Project Development with Node-RED ● Developing IoT projects using Node-RED and sensors, actuators, and communication protocols	8 hours
Pedagogy	Assignments / tutorials / peer-learning / troubleshooting/ case studies	
References/ Readings	1. Buyya, Rajkumar, and Amir Vahid Dastjerdi, eds. Internet of Things: Principles and paradigms. Elsevier, 2016. 2. Raj, Pethuru, and Anupama C. Raman. The Internet of Things: Enabling technologies, platforms, and use cases. CRC press, 2017. 3. "Internet of Things: A Hands-on Approach", by Arshdeep Bahga and Vijay Madisetti (Universities Press) 4. Research papers 5. Hagino, Taiji. Practical Node-RED Programming: Learn powerful visual programming techniques and best practices for the web and IoT. Packt Publishing Ltd, 2021. 6. https://cookbook.nodered.org/	
Course Outcomes	After completion of the course, the learner will be able to: 1. design some IOT-based prototypes 2. understand the various protocols and communication methods used in IoT systems, including MQTT, CoAP, and HTTP. 3. implement various protocols and communication methods used in IoT systems, including MQTT in NodeRED 4. design and build IoT systems for a variety of use cases, including smart home automation,	

(Back to top)

Name of the Programme: MCA
Course Code: CSA-621
Title of Course: Corporate Skills
Number of Credits: 4 (4L-0T-0P)
Effective from AY: 2022-23

Prerequisites for the course	Programme prerequisites	
Objectives	The course is aimed at learners to gain practical and essential skills to work effectively in the industry.	
Content	Understanding the Industry and Companies ● Understanding the evolution of the industry and technology and methods used ● Understanding Innovation and how new Impactful ideas have evolved ● Types of companies and typical organization - Who does What ● Understanding companies - Domain, Offering, Customers, Strategy ● Company Culture & Professionalism ● Understanding companies financially	8 hours
	Understanding Execution and day to day work in organizations ● Product Solutioning and Development - Understanding beyond the theory ● Product Management - Understanding beyond the theory ● Quality - Understanding beyond the theory ● Solutioning and Design - A key step between requirements and delivery ● Site Reliability, Devops, Support - Understanding beyond the theory ● Common Metrics and Measurements ● Key Tools in a Product Life Cycle ● Issues Management and Lifecycle - A key aspect of customer Satisfaction ● Software delivery models and Release cycles - how they work in the real world ● Usability by end user - UI/UX and other key concepts and its importance ● Understanding Data engineering and Data science ● Writing good product or service specifications which can be translated to building a good product ● Understanding data from collection to modelling to usage ● How to do effective product, competition or technical research and use it effectively ● testing and testing automation - understand beyond the theory ● what is effective program management and scrum management ● Designing for performance, scalability and reliability in products ● Effective root cause analysis and building products which can allow quicker RCA ● Understanding dev ops and its importance and role in a company ● Understanding product architecture with respect to a monolith or modularity and its pros and cons ● Governance, alerts and monitoring and its importance	20 hours
	Useful skills to work effectively in an organization ● Continuous learning and improvement - An essential skill ● Ownership and Leadership ● Analyzing one’s career path and making educated judgements ● Time management and multi-tasking model	20 hours

	● Being an effective Mentee and Mentor ● Being Inquisitive: Why is asking questions more difficult than giving answers? ● Effective Articulation and Communication ● Introducing yourself & Making Effective Presentations ● Problem breakdown and resolving model ● Effective project Management ● Mind Mapping - A powerful technique to learn ● Must have tips to succeed in any career	
	Mini-Project	12 hours
Pedagogy	Hands-on assignments / tutorials / peer-teaching / mini-project / case studies	
References/ Readings	All the course material is based on real life industry practices, experiences and case studies and focusedon application of skills and knowledge. The course is being imparted by experienced industry professionals who are still working in the industry and leading critical functions and teams and have the pedigree of building products, managing and delivering to customers, managing teams, entrepreneurs or being part of core teams in software product or services organization.	
Course Outcomes	At the end of the course, the students will be able to - 1. understand core concepts. (To measure this outcome, Question and Answers, Situations analysis, case studies would be used) 2. analyze the problem and apply the appropriate concept. (To measure this outcome, Projects and Case studies would be used) 3. give reasoning. (To measure this outcome, Problem analysis and solving techniques would be taught and used, Question and answers and use cases would be utilized) 4. apply core concepts to new situations. (To measure this outcome, Group projects and Case studies based homework would be used)	

(Back to top)