MCA (2 years) - Programme Structure

SEMESTER-1						SEMESTER-2					
Course_Name		**Contact_Hours/week**			**Credits**	**Course_Name**		**Contact_Hours/ week**			**Credits**
		L	**T**	**P**				**L**	**T**	**P**	
CSC-101	*Data Structures & Algorithms*	*3*	*0*	*0*	*3*	*CSC-201*	*Web Development*	*3*	*0*	*0*	*3*
CSC-102	*Object Oriented Concepts*	*2*	*0*	*0*	*2*	*CSC-202*	*Database Management Systems*	*3*	*0*	*0*	*3*
CSC-103	*Operating Systems*	*4*	*0*	*0*	*4*	*CSC-203*	*Mathematics for Computer Science*	*4*	*0*	*0*	*4*
CSC-104	*Internet Technologies*	*3*	*0*	*0*	*3*	*CSC-204*	*Software Design & Engineering LAB*	*0*	*2*	*4*	*4*
CSC-105	*Data Structures & Algorithms LAB*	*1*	*0*	*4*	*3*	*CSC-205*	*Web Development LAB*	*1*	*0*	*4*	*3*
CSC-106	*Object Oriented Programming LAB*	*1*	*0*	*4*	*3*	*CSC-206*	*Database Management Systems LAB*	*1*	*0*	*4*	*3*
CSC-107	*LINUX LAB*	*2*	*0*	*4*	*4*		*Elective - 1*	*4*	*0*	*0*	*4*
CSC-108	*Communication Skills*	*2*	*0*	*0*	*2*			**Total_Credits**			**24**
		Total_Credits			**24**						

SEMESTER-3						SEMESTER - 4					
Course_Name		**Contact_Hours/week**			**Credits**	**Course_Name**					**Credits**
		L	**T**	**P**							
CSC-301	*Machine Learning*	*3*	*0*	*0*	*3*	*Industry Internship/ Project*					*24*
CSC-302	*Modern Development Platforms*	*3*	*0*	*0*	*3*			**Total_Credits**			**24**
CSC-303	*Machine Learning LAB*	*1*	*0*	*4*	*3*						
CSC-304	*Modern Development Platforms LAB*	*1*	*0*	*4*	*3*						
	Elective - 2	*4*	*0*	*0*	*4*	**Total Credits**	**72+24 = 96**				
	Elective - 3	*4*	*0*	*0*	*4*						
	Elective - 4	*4*	*0*	*0*	*4*						
		Total_Credits			**24**						

Programme: MCA
Course Code: CSC-101 **Title of Course:** Data Structures & Algorithms
Number of Credits: 3 (3L-0T-0P) **Contact Hours:** 36 hours (36L-0T-0P)
Effective from AY: 2021-22

Prerequisites for the course	Program Prerequisites	
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 DefinedSelection Constructs, Repetition Constructs, Recursion Pointers Algorithm Representation: - Pseudocode and flowcharts Three level Approach Abstract Data Types (ADTs) Basic Linear Data Structures (LinkedList, Stack, Queue)	6 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	5 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	12 hours

	Divide & Conquer Strategy Algorithms based on Divide and Conquer Strategy: Sorting Algorithms (QuickSort, MergeSort) Binary Search	3 hours

	Greedy Algorithms 1. Huffman Coding Algorithm 2. Minimum Cost Spanning Tree (Prim's, Kruskal's) 3. Single Source Shortest Path (Dijkstra's)	4 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	
Learning 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 solution to a given problem. • Be able to analyze the complexity of a given algorithm	

Programme: MCA
Course code: CSC-102 **Title of course:** Object Oriented Concepts
Number of credits: 2 (2L-0T-0P) **Total contact hours:** 24 hours (24L-0T-0P)
Effective from AY: 2021-22

Prerequisites for the course	Program Prerequisites	
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	7 hours
	Object oriented principles Encapsulation Inheritance; types of inheritance; diamond problem Abstraction; virtual methods Polymorphism; overloading and overriding	6 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++)	6 hours
	Advanced features Persistence & Serialization; JSON User packages & custom libraries; reflection Predicates & streams Lambda functions	5 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	
Learning Outcomes	1. Learner will appreciate mapping real-world scenarios in the 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	

Programme: MCA
Course code: CSC-103 **Title of course:** Operating Systems
Number of credits: 4(4L-0T-0P) **Total contact hours:** 48 (48L-0T-0P)
Effective from AY: 2021-22

Prerequisites for the course	Programme Prerequisites	
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	4 hours
	Process Management Process-Concept and states, Process Creation and Control, Scheduling Criteria, Scheduling Algorithms, MultiLevel Queues., Multiple-processor scheduling, real time CPU scheduling	6 hours
	Threads Motivation and Challenges, Multithreading Models, Threading Issues, Thread libraries, Thread scheduling	4 hours
	Process Synchronization Cooperating processes and Race Conditions, The critical-section problem, Peterson's solution, mutex locks, Synchronization Hardware, Semaphores and theirImplementation, Classic problems of synchronization	6 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	4 hours
	Memory Management Hardware Support, Address Binding, Swapping, Contiguous Memory Allocation, Fragmentation, memory Protection, Paging, Structure of the page table, Segmentation, Example: Intel architecture	6 hours
	Virtual-Memory Management Background, Demand Paging, Copy-on-write, Page Replacement algorithms, Allocation of Frames, Thrashing, Allocating Kernel Memory	6 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	6 hours
	Secondary-storage Structure Overview of Mass-storage Structure, Disk Structure, Disk Attachment, Disk Scheduling ,Disk Management, Swap-Space Management	3 hours
Pedagogy	lectures/ tutorials/assignments/class presentations and debates/peer reviews/self-study.	
References/ Readings	**Main Reading** 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	
Learning 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	

Programme: M.C.A

Course Code: CSC-104 **Title of the Course:** Internet Technologies

Number of Credits: 3 (3L-0T-0P) **Contact Hours:** 36 hours (36L-0T-0P)

Effective from AY: 2021-22

Prerequisites for the course:	Programme requisites	
Objectives:	The objective 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 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.	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.	6 hours
	Network layer: Packet switching: virtual circuit & datagram networks, The Internet Protocol (IP): Forwarding and Addressing in the Internet, route aggregation, subnetting, CIDR, IP datagram, fragmentation, NAT, DHCP, ICMP. Routing protocols: shortest path, link state routing algorithm, distance vector routing. Internet routing: Autonomous Systems (AS), RIP, OSPF, BGP. Address Resolution Protocol (ARP) and RARP.	10 hours
	Internet Security protocols Basic cryptography concepts, Secure Socket Layer (SSL), Internet Security Protocol (IPSec), Virtual Private Network (VPN).	6 hours
Pedagogy:	lectures/ tutorials/assignments/self-study	

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.
Learning 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 the routing algorithms.

Programme: MCA
Course Code: CSC-105 **Title of Course:** Data Structures & Algorithms Lab
Number of Credits: 3 (1L-0T-2P) **Contact Hours:** 60 hours (12L-0T-48P)
Effective from AY: 2021-22

Prerequisites for the course	Programme Prerequisites	
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 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 Divide & Conquer Strategy MergeSort QuickSort Binary Search Algorithm Greedy Algorithms Huffman Coding Algorithm Prims' and Kruskal's Algorithm Dijkstra's Algorithm Dynamic Programming Coin Change Problem Longest Common Subsequence Floyd-Warshall Algorithm A Mini Project	2L + 8P 4L+16P 2L+8P 2L+8P 2L+8P
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	
Learning 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 solution to a given problem.	

Programme: MCA
Course code: CSC-106 **Title of course:** Object Oriented Programming Lab
Number of credits: 3 (1L-0T-4P) **Total contact hours:** 60 hours (12L-0T-48P)
Effective from AY: 2021-22

Prerequisites for the course	Program Prerequisites	
Objectives	To impart programming skills using object oriented paradigms.	
Content	**Lab assignments using Java/C++/C#** Classes and objects Class, object, 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 Object oriented principles Encapsulation Inheritance; types of inheritance; diamond problem Abstraction; virtual methods Polymorphism; overloading and overriding Object oriented features Interfaces Access modifiers Errors & Exceptions; user-defined exceptions Collections Anonymous & Inner classes Type parametric polymorphism Advanced features Persistence & Serialization; JSON User packages & custom libraries; reflection Predicates & streams Lambda functions Mini-Project	3L+12P 3L+12P 3L+12P 3L+12P
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	
Learning 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	

Programme: M.C.A

Course Code: CSC-106 **Title of the Course:** LINUX Lab

Number of Credits: 4 (2L-0T-2P) **Contact hours:** 72 hours (24L-0T-48P)

Effective from AY: 2021-22

Prerequisites for the course:	Program Prerequisites	
Objectives:	The objective is to introduce students to the Linux operating system environment and provide a 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 (<,<<, >, >>)	3L + 3P
	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	4L + 8P

	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.	6L + 8P
	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	1L + 2P
	Package management: Installing & removing packages	1L+1P
	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.	4L + 8P
	System programming Introduction to system programming, System calls and library functions. **Files and Directory system calls** List of sample programs *1.* Write a program to implement the functionality of Linux command *touch* *2.* Write a program to implement the functionality of Linux command *cat*	5L + 18P

	3. Write a program to implement the functionality of Linux command *ls* 4. Write a program to redirect the output of all the printf statements to a user file using dup system call. 5. Write a program to read the standard input from a user file using dup system call. 6. Write a program to implement the functionality of Linux command *chmod* 7. Write two programs : one called parent.c, the other called child.c. The parent program reads two integers from the keyboard and arithmetic operator (+, -, *, /). The read information is transmitted to a child process. After the child process finishes the operation, it transmits the result to the parent process. The parent process prints the result on the screen. 8. Write a c program namely "parent.c", which reads the processes along with their burst time (bt) and saves it in a file. Using fork, create a child process namely fcfs.c, which takes the filename containing process information as a parameter from the parent. The child process task is to calculate the average waiting time using the FCFS scheduling algorithm.	
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. C and Unix Programming – Kerningham and Pike, Prentice Hall 4. UNIX man pages	
Learning 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	

Programme: M.C.A

Course Code: CSC-108 **Title of the Course:** Communication Skills

Number of Credits: 2 (2L-0T-0P) **Contact Hours:** 24 hours (24L-0T-0P)

Effective from AY: 2021-22

Prerequisites for the course:	Programme requisites	
Objectives:	To introduce essentials of effective communication in different contexts	
Content:	**Oral Communication** Difference between formal and informal communication, importance of non verbal communication, skills required for effective communication, Public Speaking Skills.	4 hours
	Written Communication Writing cover letters, Resumés/CVs/Biodata, Letters of Invitation, Report Writing	4 hours
	Content creation Creating content for the website, creating profiles, creating content for brochures of events.	4 hours
	Multimedia and E-Correspondence Conducting Research before presentation, Making PowerPoint Presentation effective (visual), Communication during PowerPoint Presentation, Email etiquette (components, formats, attachments, content and language), Maintaining social media presence.	6 hours
	Preparing for Interview Types (personal, telephonic, online), Techniques of answering interviews, Participating in group discussions.	4 hours
	Allied Communication Effective Reading techniques, analyzing feedback and giving inputs.	2 hours
Pedagogy:	Lectures/ tutorials/laboratory work/ field work/outreach activities/ project work/ vocational training/viva/ seminars/ term papers/assignments/presentations/ self-study/ Case Studies etc. or a combination of some of these. Sessions shall be interactive in nature to enable peer group learning.	
References/ Readings	1. Kelly M. Quintanilla and Shawn T. Wahl, "Business and Professional Communication " Sage Publications, 2018,	

	2. Anjanee Sethi ,Bhavna Adhikari, "Effective Business Communication " Tata MacGraw Hill Education, India. 2009; 3. Nido Qubein, "How to be a Great Communicator in Person, On Paper, and on Podium" Viva Books, India. 2008; 4. Stanton, Nicky. "Mastering Communication", (5th Edition), Macmillan, 2009. 5. Dalmar, Fisher. "Communication in Organisation", West Pub, 1993. 6. Kilian, Crawford. "Writing for the Web. Self-Counsel Press", Fifth edition, 2015. 7. Kallos, Judith. "Email Etiquette Made Easy", Lulu.com. 2007.
Learning Outcomes	The participant will be able to facilitate interpersonal Communication, participate in group discussions, and to write effectively.