Graphic Era
Deemed to be University

Program Scheme

Master of Computer Applications Degree Program

Program Code: OMC

Directorate of Distance and Online Education

Batch 2023-2025

University's Vision, Mission, and Core Values

Vision: We visualize Graphic Era (Deemed to be University) as an internationally recognized, equity-driven, ethically engaged, diverse community whose members work collaboratively for positive transformation in the world, through leadership in teaching, research, and social action.

Mission: The mission of the university is to promote learning in true spirit and offer knowledge and skills in order to succeed as professionals. The university aims to distinguish itself as a diverse, socially responsible learning community with high-quality scholarship and academic rigor

Core Values:

- Continuous learning and improvement
- Simplicity
- Integrity and trust
- Ethics

Program Scheme: Master of Computer Applications

1. **Title of the Degree:** Master of Computer Applications (MCA)
2. **Mode of Study:** Fully Online
3. **Program Curriculum will be Effective from:** Academic Year 2023-2024
4. **Rationale for the Program:**

Computers, computer networks, and mobile communication have ushered in the digital revolution in the recent past. The fast-growing information and communication technology (ICT) is critical to strategic planning in most business houses, government organizations, and educational institutes all over the world. Organizations that strive to leverage the latest ICT tools require expert professionals who can apply the principles of computer science and information technology to address the issues effectively. To meet the shortage of qualified professionals in the IT industry, Graphic Era Deemed to be University has designed this Master of Computer Applications (MCA) degree program. The broad objective of this postgraduate program is to prepare graduates for productive careers in the software industry and academia. To accomplish these objectives, the university provides an outstanding environment for teaching and research in the core and emerging areas of this discipline.

The program lays immense emphasis on giving the students a thorough and sound background in theoretical and application-oriented courses relevant to the latest ICT paradigm. The program also focuses on the application of software technology to solve mathematical, computing, communications, networking, and commercial problems.

Professionals with an MCA degree are sought after in numerous corporate sectors, such as IT, Medical Sciences, and Engineering. These sectors need personnel having advanced knowledge of the application of computers to solve real-life problems. Several technology conglomerates in India have job openings for such candidates. With the right amount of experience and skillset, MCA candidates can find several challenging and rewarding career opportunities.

5. Program Educational Objectives (PEOs)

The educational objectives of the MCA program are to:

PEO 1. Empower students with employability towards building successful careers based on a sound understanding of theoretical and applied aspects and methodology to solve multidisciplinary real-life problems.

PEO 2. Develop professional graduates ready to work with a sense of responsibility and ethics.

PEO 3. Instil competency to pursue higher studies and research in areas of computer applications and other professionally related fields.

PEO 4. Inculcate the ability to adapt to changing technology through continuous learning.

6. Programme Outcomes (POs)

Serial Number	Graduate Attribute Theme	The Complete PO Statement
		After the successful completion of the MCA program, the graduates will be able to:
PO-1.	Knowledge Application	Apply the knowledge of mathematics, management, and computer applications to the solution of complex real-world problems.
PO-2.	Problem Analysis	Identify, formulate, review, and analyze complex problems reaching substantiated conclusions using principles of mathematics, management sciences, and computer applications.
PO-3.	Design/Development of Solutions	Design solutions for complex real-world problems and design system components or processes that meet the specified needs with appropriate consideration for health and safety, and cultural, societal, and environmental considerations.
PO-4.	Investigations of Complex Computing Problems	Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO-5.	Modern Tool Usage	Create, select, and apply appropriate techniques, resources, and modern computer software and IT tools including prediction and modeling to complex software engineering activities with an understanding of the limitations.
PO-6.	Environment and Sustainability	Understand the impact of professional software engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO-7.	Ethics	Apply ethical principles and commit to professional ethics and responsibilities and norms of the development practice.
PO-8.	Individual and Teamwork	Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO-9.	Communication	Communicate effectively on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO-10.	Project Management and Finance	Demonstrate knowledge and understanding of the software engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO-11.	Life-Long Learning	Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.
PO-12.	Innovation and Entrepreneurship	Identify a timely opportunity and use innovation to pursue that opportunity to create value and wealth for the betterment of the individual and society at large.

7. Programme Specific Outcomes (PSOs)

At the end of the MCA program, the graduate will be able to:

PSO 1. Apply fundamental principles and methods of Computer Science to a wide range of applications.

PSO 2. Design, implement, and document solutions to significant computational problems.

PSO 3. Demonstrate an understanding of the basics of computer applications.

PSO 4. Engage in continued professional development in a career in computer applications.

8. Program Structure:

Semester 1				
Sl. No.	**Course Type***	**Course Code**	**Course Title**	**Credits**
A	Audit	23OMC100A	Fundamentals of Computers	0
B	Bridge	23OMC100B	Introduction to Operating Systems	0
C	Bridge	23OMC100C	Mathematical Foundation of Computer Science	0
1	DSC	23OMC101	Full Stack Development	3
2	DSC	23OMC102	Computer Networks	3
3	DSC	23OMC103	Programming and Problem-Solving	3
4	DSC	23OMC104	Advanced Operating Systems	3
5	DSE	**Discipline-Specific Elective - 1**		3
		23OMC105A	Advanced Computer Organization	
		23OMC105B	Green Computing	
		23OMC105C	Discrete Structures and Combinatorics	
		23OMC105D	Cloud Computing	
6	SEC	23OMC106	Career Skills	1
7	VAC	23OMC107	General Proficiency/NCC/Seminar/Research/Yoga*	1
8	DSC	23OMC108	Full Stack Development Laboratory	2
9	DSC	23OMC109	Operating Systems and Computer Networks Laboratory	2
10	DSC	23OMC110	Programming and Problem-Solving Laboratory	2
			Total Credits Over the Semester	**23**

*DSC: **D**iscipline-**S**pecific Core **C**ourse SEC: **S**kill-**E**nhancement **C**ourse	DSE: **D**iscipline-**S**pecific **E**lective Course VAC: **V**alue **A**ddition **C**ourse	GE: **G**eneral **E**lective Course

Semester 2				
Sl. No.	**Course Type***	**Course Code**	**Course Title**	**Credits**
A	Bridge	23OMC200A	Introduction to Database Management Systems	0
B	Bridge	23OMC200B	Introduction to Object-Oriented Programming	0
1	DSC	23OMC201	Advanced Database Management Systems	3
2	DSC	23OMC202	Advanced Java Programming	3
3	DSC	23OMC203	Advanced Data Structures	3
4	**DSE**	**Discipline-Specific Elective - 2**		3
		23OMC204A	Data Mining and Warehousing	
		23OMC204B	Python Programming	
		23OMC204C	Software Project Management	
		23OMC204D	Probability and Statistics	
5	**GE**	**General Elective - 1**		2
		23OMC205A	Research Methodology	
		23OMC205B	Entrepreneurship	
6	SEC	23OMC206	Career Skills	2
7	SEC	23OMC207	Mini Project/Research Publication	1
8	DSC	23OMC208	Advanced Database Management Systems Laboratory	2
9	DSC	23OMC209	Advanced Java Programming Laboratory	2
10	DSC	23OMC210	Advanced Data Structures Laboratory	2
			Total Credits Over the Semester	**23**

Semester 3				
Sl. No.	**Course Type***	**Course Code**	**Course Title**	**Credits**
A	Audit	23OMC300A	Competitive Programming	0
B	Bridge	23OMC300B	Introduction to Software Engineering	0
1	DSC	23OMC301	Design and Analysis of Algorithms	3
2	DSC	23OMC302	Mobile Application Development	3
3	DSC	23OMC303	Artificial Intelligence and Machine Learning	3
4	**DSE**	**Discipline-Specific Elective - 3**		3
		23OMC304A	Software Testing and Quality Assurance	
		23OMC304B	Human-Computer Interaction	
		23OMC304C	Theory of Computation and Compiler Construction	
		23OMC304D	Operations Research	
5	SEC	23OMC305	Career Skills	2
6	SEC	23OMC306	Mini Project/Research Seminar	2
7	DSC	23OMC307	Design and Analysis of Algorithms Laboratory	2
8	DSC	23OMC308	Mobile Application Development Laboratory	2
9	DSC	23OMC309	Artificial Intelligence and Machine Learning Laboratory	2
			Total Credits Over the Semester	**22**

Semester 4				
Sl. No.	**Course Type***	**Course Code**	**Course Title**	**Credits**
1	DSC	23OMC401	Data Science using R	3
2	**DSE**	**Discipline-Specific Elective - 4**		**3**
		23OMC402A	Cryptography	
		23OMC402B	Cybersecurity/Information Security	
		23OMC402C	University-Approved MOOC or Certification	
		23OMC402D	Computer-Aided Simulation and Modelling	
3	**DSE**	**Discipline-Specific Elective - 5**		**3**
		23OMC403A	C# and .NET	
		23OMC403B	Advanced Graphics and Visual Computing	
		23OMC403C	Soft Computing	
		23OMC403D	Internet of Things	
4	**GE**	**General Elective - 2**		**3**
		23OMC404A	Personal Finance	
		23OMC404B	Digital Marketing	
5	SEC	23OMC405	Internship/Dissertation/Capstone Project	8
6	DSC	23OMC406	Data Science Laboratory	2
			Total Credits Over the Semester	**22**
			Total Credits Over the Program	**90**

9. Programme Articulation Matrix (Course-PO-PSO Map)

Sem.	Course Title	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
1	Fundamentals of Computers																
1	Introduction to Operating Systems																
1	Mathematical Foundation of Computer Science																
1	Full Stack Development																
1	Computer Networks																
1	Programming and Problem-Solving																
1	Advanced Operating Systems																
1	Advanced Computer Organization																
1	Green Computing																
1	Discrete Structures and Combinatorics																
1	Cloud Computing																
1	Career Skills																
1	General Proficiency/ NCC/ Seminar/Research/Yoga*																
1	Full Stack Development Laboratory																
1	Operating Systems and Computer Networks Laboratory																
1	Programming and Problem-Solving Laboratory																
2	Introduction to Database Management Systems																
2	Introduction to Object-Oriented Programming																
2	Advanced Database Management Systems																
2	Advanced Java Programming																
2	Advanced Data Structures																
2	Data Mining and Warehousing																
2	Python Programming																
2	Software Project Management																
2	Probability and Statistics																
2	Research Methodology																
2	Entrepreneurship																
2	Career Skills																
2	Mini Project/Research Publication																
2	Advanced Database Management Systems Laboratory																
2	Advanced Java Programming Laboratory																
2	Advanced Data Structures Laboratory																

Sem.	Course Title	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
3	Competitive Programming																
3	Introduction to Software Engineering																
3	Design and Analysis of Algorithms																
3	Mobile Application Development																
3	Artificial Intelligence and Machine Learning																
3	Software Testing and Quality Assurance																
3	Human-Computer Interaction																
3	Theory of Computation and Compiler Construction																
3	Operations Research																
3	Career Skills																
3	Mini Project/Research Seminar																
3	Design and Analysis of Algorithms Laboratory																
3	Mobile Application Development Laboratory																
3	Artificial Intelligence and Machine Learning Laboratory																
4	Data Science using R																
4	Cryptography																
4	Cybersecurity/Information Security																
4	University-Approved MOOC or Certification*																
4	Computer-Aided Simulation and Modelling																
4	C# and .NET																
4	Advanced Graphics and Visual Computing																
4	Soft Computing																
4	Internet of Things																
4	Personal Finance																
4	Digital Marketing																
4	Internship/Dissertation/ Capstone Project																
4	Data Science Laboratory																
* CO-PO-PSO mapping depends on the course/certification chosen by the student.																	

10. **Programme Regulations:** The regulations guiding this programme are available in the Program Guide.

MCA 1st Semester – Full Stack Development

Program	Master of Computer Applications
Semester	1
Course Title	Full Stack Development
Course Code	OMC101
Course Credits	3
Course Type	Core

1. Course Summary

The aim of this course is to gain the skills and knowledge necessary to build simple web applications as well as full-stack web applications using modern and scalable web technologies and increase employability as a full-stack developer. The students are taught the basics of HTML, CSS, JavaScript, PHP, and the basic components of Full Stack development using MERN stack widely used in the industry for developing web pages. Students will learn the use of XHTML and CSS for developing presentable web pages. They will also be able to create dynamic web pages by applying event-handling mechanisms using JavaScript. Students will understand the concepts of cookies and sessions in PHP for creating large web applications. Students will learn the MERN stack that consists of four technologies – MongoDB: a database used to store data in JSON documents, Express.js: a framework for Node.js that provides features for developing web applications, ReactJS – a JavaScript library for building user interfaces and Node.js: an open-source cross-platform JavaScript runtime environment that executes JavaScript code outside of a web browser. Students are trained to create interactive web pages using ReactJS, Node.js, MongoDB, and PHP.

Course Outcomes (COs)

After the successful completion of this course, the student will be able to:

- **CO-1.** Describe the usage of XHTML tags, and CSS for developing presentable web pages.[L-1]
- **CO-2.** Develop dynamic web pages by applying event-handling mechanisms using JavaScript.[L-5]
- **CO-3.** Develop web pages using cookies, sessions, and databases in PHP.[L-5]
- **CO-4.** Describe the concepts of MERN stack used in Full Stack Development and the features and components of ReactJS[L-1]

2. Course Contents

Sr. No	Units	Unit Outcomes After the successful completion of the unit, the learner should be able to:
1	**Unit 1: Basics of XHTML** • Standard Structure of XHTML document • Difference between HTML and XHTML • Basic Text Markup Elements – o paragraph <p> , heading tags H1..H6 o break, italic, bold, superscript, subscript, emphasis, <hr> • Character entities • Hyperlinks • Images • Lists , , <dl> • tables <table> • division and span • Form Elements o Label o Text boxes - password, button, submit, reset o Placeholder o Textarea o Radio button o Checkboxes o Button o Dropdown <select>	Demonstrate the use of various XHTML tags for creating web pages.
2	**Unit 2 – CSS** • Introduction to CSS • CSS Properties – text, font, list, color • Selectors o Simple/Element Selector o Id Selector o Class Selector o Generic Selector o Universal Selector o Pseudo-class Selector • Types and Levels of Style o Inline o Internal o External • CSS Box Model • Background images • Examples using HTML and CSS	Demonstrate the use of cascading style sheets for creating attractive web pages.
3	**Unit 3: Introduction to JavaScript, Event Handling, DOM, and Dynamic Documents with JavaScript** **Introduction to JavaScript -** • Overview of JavaScript • JavaScript Syntax • Variables and Data Types • Screen Output and Keyboard Input	Explain the basic concepts of Java Script, demonstrate and implement pattern matching concepts and create interactive web pages Explain and demonstrate the use of DOM for creating interactive and dynamic event-handling web pages using JavaScript.

	○ alert() ○ confirm() ○ prompt() ○ document.write() • Controls and Loops • String Methods • Arrays • Functions • Pattern Matching **Event Handling, DOM, and Dynamic Documents with JavaScript -** • Document Object Model (DOM) • DOM tree structure of HTML • Accessing elements in JavaScript • Event and Event Handling ○ Handling events from body elements ○ Handling events from button elements ○ Handling events from text boxes • Validations on Forms simple example • Changing colors and fonts	
4	**Unit 4: PHP Basics and Advance Features** **PHP Basics –** • Introduction and basic syntax of PHP, • Control Statements with examples • Output Statements • String Functions • Arrays • Functions **PHP Advance Features –** • Form Handling • Cookies and Sessions Management ○ Creating cookies and session variables in PHP programs • Creating a simple database and database operations	Explain the basics concepts of PHP Explain and demonstrate the usage of various built-in functions of string and arrays in PHP Discuss the use of cookies and sessions in PHP
5	**Unit 5: Introduction to MERN** **Introduction to MERN -** • Overview of Full Stack Web Development • Overview of MERN • Overview of MERN Components ○ ReactJS ○ Node.js ○ Express ○ MongoDB • Tools and Libraries • React library **Introduction to ReactJS -** • React features, benefits, and applications	Explain the basics of full-stack web development. Discuss the MERN stack and its components. Describe the features and benefits of using ReactJS an open-source JavaScript library Create and execute a simple ReactJS application.

	• Advantages and disadvantages of ReactJS • ReactJS Vs other Front-End Technologies • ReactJS development environment setup • Creating and executing a new ReactJS project • Folder Structure of ReactJS applications	

3. Course Articulation Matrix (CO-PO-PSO Map)

	Programme Outcomes (POs)												Programme Specific Outcomes (PSOs)			
	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	PSO-3	PSO-4
CO-1	3	2	3		3			2		1	2		3	3		
CO-2	3	2	3		3			2		1			3	3		
CO-3	3	3	3		3	2		2		1		2	3	3	3	2
CO-4	3	3	3		3	2		2	2	1		2	3	3	3	3
	3: Very Strong Contribution, 2: Strong Contribution, 1: Moderate Contribution															

4. Course Resources

a. Essential Reading

1. Robert W. Sebesta, "Programming the world wide web", 6th edition, Pearson education.
2. Vasan Subramanian, "Pro MERN Stack, Full Stack Web App Development with Mongo, Express, React and Node", 2nd Edition, Apress.

b. Recommended Reading

1. Kogent Learning Solutions Inc., "HTML 5: Covers CSS3, JavaScript, XML, XHTML, AJAX, PHP & jQuery: Black Book", Dreamtech Press.
2. Eddy Wilson, 2018, "MERN Quick start guide: Build Web applications with MongoDB, Express.js, React and Node", Packt publishing.

c. Magazines and Journals

1. IEEE Transaction on Computers
2. International Journal of Computer Science

d. Websites

1. https://www.coursera.org/
2. http://nptel.ac.in/

e. Other Resources

1. https://ocw.mit.edu/index.htm
2. Course Video Lectures on ILearn, BrightSpace

--**--

OMC102 – Computer Networks

Program	Master of Computer Applications
Semester	1
Course Title	Computer Networks
Course Code	OMC102

Course Credits	3
Course Type	Core Course

1. Course Summary

This course deals with the concept and technologies used in modern computer networking and data communication. A computer network interconnects two or more computing devices. Since implementing computer networking software is a highly complex task, it has been implemented in different layers. Every layer has a well-defined service to perform. This course facilitates the students to understand the function of different layers and IEEE standards employed in computer networking. The students are taught the methods to enhance network performance such as routing and congestion control. Fundamental concepts of computer networks, different network models, and topologies are covered.

2. Course Outcomes (COs)

After the successful completion of this course, the student will be able to:

- **CO-1.** Explain the principles, mechanisms, and functionalities of network applications, transport layer protocols, network layer design, and link layer services in computer networks.(L2)
- **CO-2.** Illustrate the principles and architectures of network applications, including protocols such as HTTP, FTP, SMTP, POP3, IMAP, and DNS.(L2)
- **CO-3.** Demonstrate use of different computer network components, including hardware, media, and topologies.(L3)
- **CO-4.** Compare error-detection and correction techniques, multiple access protocols, and Ethernet technologies in the context of link layer services and local area networks.(L4)
- **CO-5.** Evaluate routing algorithms (such as link-state and distance vector) in terms of their efficiency, scalability, and adaptability to various network environments. (L5)

3. Course Contents

Sr. No	Units
1	**Introduction:** Data Communication Basics, History of Computer Networking and the Internet. Internet, Protocol, Services. Computer Network: Hardware, Media and topology. **Protocol layering**: The OSI Reference Model and the TCP/IP protocol stack. Internet Access Networks. Circuit and Packet Switching, Delays: Processing, Queuing, Transmission and Propagation delays.
2	**Application Layer:** Principles and Architectures of Network Applications. Application Layer Protocols- The Web and http: Persistent and Non-persistent connections, http message format, cookies, proxy server, conditional GET, File Transfer Protocol. Email: SMTP, mail message formats, mail access protocols: POP3, IMAP, MIME. **DNS:** Services, how it works, Root, Top-Level and Authoritative DNS servers, Resource Records, DNS messages. A simple Introduction to p2p files distribution: Bit Torrent

3	**Transport Layer**: Introduction and Services, Transport layer in internet, Difference between Connection Oriented and Connectionless services. UDP: Segment structure, checksum in UDP. **TCP**: the principles behind connection-oriented data transfer, stop-and-wait, Go Back N, Selective Repeat. Connection Establishment, TCP header, Round Trip Time, designing a reliable data transfer protocol.
4	**Network Layer**: Network Layer Design Issues, Packet Forwarding and Routing, Difference between Virtual Circuits and Datagram networks, The Internet Protocol (IP), Datagram format, IP fragmentation, IPv4 addressing, subnets, CIDR, classful addressing, DHCP, Network Address Translation (NAT). IPv6 Header, Moving from IPv4 to IPv6: tunneling, dual stack and header translation. **Routing Algorithms**: Link state (LS), Distance Vector (DV). Routing in the Internet: RIP, OSPF & BGP.
5	**Link Layer and Local Area Network**: **Introduction and Services**: Service provided by the LL, Implemented. Error-Detection and Correction Techniques: Parity checks, Check-summing methods, Cyclic Redundancy Check (CRC). Multiple Access protocols: Channel partitioning, Random access. **Ethernet**: Frame structure, CSMA/CD, Ethernet technologies. **Signals-** analog and digital signals, periodic and a periodic signal, Digital Data Conversion: unipolar, polar, bipolar. **Analog data conversion**: - PAM, PCM, sampling. Modulation techniques: - ASK, FSK, PSK, AM, FM, PM.

4. Course Articulation Matrix (CO-PO-PSO Map)

	Programme Outcomes (POs)												**Programme Specific Outcomes (PSOs)**			
	PO-1	**PO-2**	**PO-3**	**PO-4**	**PO-5**	**PO-6**	**PO-7**	**PO-8**	**PO-9**	**PO-10**	**PO-11**	**PO-12**	**PSO-1**	**PSO-2**	**PSO-3**	**PSO-4**
CO-1	3										2		3		1	1
CO-2	3										2		3		2	1
CO-3	3	2									2		3		2	1
CO-4	3	3		2							2		3	2	2	2
CO-5	3	3	2	2	2						2		3	2	2	3
	3: Very Strong Contribution, 2: Strong Contribution, 1: Moderate Contribution															

5. Course Resources

a. Essential Reading

1. Course Self-Learning Material
2. James F. Kurose and Keith W. Ross, 2017, Computer Networking: A Top-Down Approach, 7th Edition,Prentice Hall.

b. Recommended Reading

1. Andrew S. Tanenbaum and David J. Wetherall, 2014, Computer Networks, 5th Edition, Pearson
2. Computer Networks: A Top-Down Approach by Behrouz A. Forouzan and Firouz Mosharraf.New York, NY: McGraw-Hill, 2012.

c. Websites

1. https://www.coursera.org/
2. http://nptel.ac.in/

MCA 1st Semester – Programming and Problem-Solving

Program	Master of Computer Applications
Semester	1
Course Title	Programming and Problem-Solving
Course Code	OMC103
Course Credits	3
Course Type	Core Theory

1. Course Summary

The aim of this course is to familiarize the students with the fundamental concepts of computational thinking, problem-solving strategies, and programming in C language. This course provides knowledge of writing C programs using arrays and strings, structures and unions, simple user-defined functions as well as recursive functions, and implementing file operations such as opening, closing, reading, and writing to files. The students are taught how to choose an appropriate C construct to solve a given problem statement.

2. Course Outcomes (COs)

After the successful completion of this course, the student will be able to:

CO-1. Describe the fundamental concepts of computational thinking and problem-solving strategies. [L-1]

CO-2. Demonstrate the use of arrays, strings, structures, and unions in the 'C' programming language. [L-3]

CO-3. Demonstrate the use of re-useable code using functions in 'C'. [L-3]

CO-4. Describe and implement file handling mechanism in 'C' programs. [L-3]

3. Course Contents

Sr. No	Units
1	**Unit 1 – Computational Thinking, Problem Solving, and Programming Concepts** **Computational Thinking -** • What is computational thinking? • Computational thinking approaches • Information and Data – Converting Information to Data • Data Types and Encoding **Problem-Solving and Programming Concepts –** • Problem-Solving techniques

	• Algorithms • Flowcharts • Pseudocode • Classification and Characteristics of programming language • Programming paradigms • Procedure-oriented programming • Object-oriented programming languages

2	**Unit 2: Introduction to C Programming, Variables and constants, Operators and Expressions, Input/Output functions in C**
	
	Introduction to 'C' Programming -
	• Characteristics of 'C'
	• Structure of C Program
	• The life cycle of the C Program
	• First C Program
	• Commands to run a C Program
	• Comments Style in 'C'
	• Programming errors -
	• Syntax error and semantic errors
	• Logical and runtime errors
	
	Variables and Constants -
	• C Character Set
	• Identifiers and keywords
	• Variables and constants
	• Escape sequence
	• Data Types
	
	Operators and Expressions in C -
	• Assignment Operators, Arithmetic Operators
	• Relational Operators, Logical Operators
	• Increment and Decrement Operators
	• Conditional Operators
	• Typecast Operators, Sizeof Operator
	• Associativity and precedence of operators
	• Evaluation of Expressions
	
	Input/Output (I/O) Functions -
	• Types of I/O Functions
	• Unformatted I/O Functions
	  ○ getchar(), putchar()
	  ○ gets(), puts()
	  ○ getch(), putch()
	• Formatted I/O functions
	  ○ Format specifiers
	  ○ scanf()
	  ○ printf()

	Unit 3: Conditional and Control Statements, Functions **Conditional and Control Statements -**

3	• Conditional Branching Statements
	  o If statement, If then else statement
	  o Nested if
	  o Switch Statement
	• Loops
	  o For loop
	  o While loop
	  o Do While loop
	• Jump Statement
	  o break
	  o continue
	  o goto
	  o return
	
	Functions -
	• Library functions
	• Function declaration and definition
	• Function prototype and call
	• Return Statement
	• Function with and without arguments
	• Function with and without return value
	• Function call by value and call by reference
	• Advantages of functions
	• Function call stack and activation records
	• Recursive functions
	• Recursive Vs Iterations
	• Examples of recursive functions
	• Static and Dynamic Linking

4	**Unit 4: Pointers, Arrays, and Strings**
	
	Pointers -
	• Pointers and their characteristics
	• Pointer declaration and assignment
	• Dereferencing pointer variables
	• Pointer arithmetic
	• Pointers and functions
	• Dynamic memory allocation – malloc(), calloc() realloc(), free() functions
	• Memory leak and segmentation fault
	• Debugging and Testing
	
	Arrays -
	• Single-dimensional array
	  o Array declaration
	  o Accessing elements of an array
	  o Initialization
	  o Array operations (insert, delete, sort, and search)
	• Two-dimensional arrays
	  o Declaration of a 2D array
	  o Initialization
	  o Operations on Matrices (addition, product, transpose)
	
	Strings -

	• Declaration and initialization of strings • Input and Output of strings • Formatting strings • String handling functions
5	**Unit 5: Structures and Unions, File Handling** **Structures and Unions –** **Structures -** • Need of structures, Declaring and defining a structure • Initialization of structure variables • Accessing structure members, assignment of structure variables • Size of a structure • Array of structures, Structure with arrays • Nested structure • Structures and functions • Structures and Pointers • Self-referential structure **Unions -** • Declaring and defining a union • Initialization and access of union variables • Size of a union • Nested unions • Difference between Structure and Union **File Handling -** • Types of files • File modes, Opening, closing, and end of a file • Character I/O functions - fputc() , fgetc() • Integer I/O functions - putw(), getw() • String I/O functions - fputs(), fgets() • Formatted I/O functions - fprintf(), fscanf() • Block Read/Write functions - fwrite(), fread() • Random access to a file - fseek(), ftell(), rewind() • Error handling in files

4. Course Articulation Matrix (CO-PO-PSO Map)

	Programme Outcomes (POs)												Programme Specific Outcomes (PSOs)			
	PO-1	**PO-2**	**PO-3**	**PO-4**	**PO-5**	**PO-6**	**PO-7**	**PO-8**	**PO-9**	**PO-10**	**PO-11**	**PO-12**	**PSO-1**	**PSO-2**	**PSO-3**	**PSO-4**
CO-1	3	2	3		3			2		1	2		3	3		
CO-2	3	2	3		3			2		1			3	3		
CO-3	3	3	3		3	2		2		1		2	3	3	3	2
CO-4	3	3	3		3	2		2	2	1		2	3	3	3	3
	3: Very Strong Contribution, 2: Strong Contribution, 1: Moderate Contribution															

5. Course Resources

a. Essential Reading

1. David D. Riley and Kenny A. Hunt, (2014), "Computational thinking for the Modern Problem Solver", Chapman & Hall/CRC.
2. Yashavant Kanetkar, (2016), "Let Us C",14th Edition, BPB Publication.

b. Recommended Reading

1. E. Balagurusamy, (2015), "Programming in ANSI C",6th Edition, McGraw-Hill.
2. Brian W Kernighan & Dennis M Ritchie, (1988), "The C Programming Language", 2nd Edition, Prentice Hall.
3. Steve Oualline,(2011), "Practical C Programming", 3rd Edition, Orielly Publishers.

c. Magazines and Journals

1. IEEE Transaction on Computers
2. International Journal of Computer Science

d. Websites

1. https://www.learn-c.org
2. https://www.programiz.com/c-programming

e. Other Resources

3. https://ocw.mit.edu/index.htm
4. Course Video Lectures on ILearn, BrightSpace

--**--

MCA104- Advanced Operating Systems

Program	Master of Computer Applications
Semester	1
Course Title	Advanced Operating Systems
Course Code	MCA 104
Course Credits	3
Course Type	Core Theory Course

1. Course Summary

This course is designed to build on the foundational knowledge acquired in the previous course by exploring advanced topics in the subject. Students will delve into resource management, including file and storage management, as well as the concepts and mechanisms related to data protection and security features of an operating system. The course covers various types of operating systems, including those developed for real-time, parallel, distributed, and cloud computing environments. A case study of two popular operating systems, Windows and Linux, is also included in the course to help students analyze their design, architecture, and functionality

2. Course Outcomes (COs)

After the successful completion of this course, the student will be able to:

CO-1. List the functions and purposes of modern operating systems. [L-1]
CO-2. Describe the file and storage management concepts. [L-2]
CO-3. Differentiate between the protection and security features of an operating system and its implementation mechanisms. [L-4]
CO-4. To interpret the concept of virtualization and describe its advantages, and need. [L-5]
CO-5. To analyze and compare the design, features, and functionality of Linux, and Windows operating systems through a case study. [L-5]

3. Course Contents

Sr. No.	Units
1	**Unit 1: 10 Hrs** **File and Storage Management** • Overview of OS Objectives and Functions • Files and File systems • File organization and Access • File Directory and Sharing • Secondary storage management **Protection and Security** • Goals of Protection • Principles of Protection • Domain of Protection • Access Control
2	**Unit 2:** **Real-Time Operating System (RTOS) (10 hrs)** • Background • Characteristics of Real-Time Operating Systems • Types of RTOS • RTOS kernel and function • Task Management • Real-Time Scheduling • Application Example
3	**Unit 3: (10 hrs)** **Distributed Systems** • Advantages of Distributed Systems • Distributed Operating Systems • Types of Distributed Operating Systems • Robustness and Design Issues of Distributed OS **Parallel Systems** • Definition, Parallel vs Distributed Systems, Example • Types of Parallel Systems: Overview
4	**Unit 4: (10 hrs)** **Virtualization** • Definition, Advantages, and Needs • Types

	• Building Block Diagram • Example **Cloud Operating Systems** • Goals and working • Examples of Cloud Operating Systems
5	**Unit 5** **(10 hrs)** **Case Studies** A comparative analysis of Linux and Windows 7 operating systems based on, • Architecture • Process Management • Memory Management • Security features

4. Course Articulation Matrix (CO-PO-PSO Map)

	Program Outcomes (POs)												**Programme Specific Outcomes (PSOs)**			
	PO-1	**PO-2**	**PO-3**	**PO-4**	**PO-5**	**PO-6**	**PO-7**	**PO-8**	**PO-9**	**PO-10**	**PO-11**	**PO-12**	**PSO-1**	**PSO-2**	**PSO-3**	**PSO-4**
CO-1	3	2								1	1	1	2	2	2	2
CO-2	3	2	1							1	1	1	2	2	2	2
CO-3	3	2	2							1	1	1	2	2	2	2
CO-4	3	3	3							1	1	1	2	2	2	2
CO-5	3	3	3							2	1	1	2	2	2	2
3: Very Strong Contribution, 2: Strong Contribution, 1: Moderate Contribution																

5. Course Resources

a. Essential Reading

1. Abraham Silberschatz, Peter B Galvin, Greg Gagne, "Operating System Concepts", Wiley India Pvt. Ltd 2018, 9th Edition
2. William Stallings, "Operating Systems Internals and Design Principles", Pearson, 2018, 9th Edition

b. Recommended Reading

1. Andrew S. Tanenbaum, "Distributed Operating Systems", Pearson 2002, 1st Edition.

c. Websites

1. https://www.coursera.org/
2. http://nptel.ac.in/

OMCXXX – Advanced Computer Organization

Program	Master of Computer Applications
Semester	1
Course Title	Advanced Computer Organization
Course Code	OMCXXX

Course Credits	3
Course Type	Elective Course

1. Course Summary

This course covers fundamental concepts and advanced topics in computer organization. It includes computer fundamentals, computer systems, the central processing unit (CPU), the processor, parallel organization, general-purpose graphic processing units (GPUs), and control unit operation. Students learn about computer evolution, performance issues, memory systems, I/O modules, instruction sets, and processor structure. They explore parallel processing, multicore computers, and GPU architecture. By the end of the course, students will have a comprehensive understanding of computer organization and be prepared to analyze and optimize computer systems.

2. Course Outcomes (COs)

After the successful completion of this course, the student will be able to:

- **CO-1.** Describe the organization and architecture of computer systems, including their components and interconnection structures.(L2)
- **CO-2.** Use Ahmdahl's Law and Little's Law to analyze and predict performance issues in computer systems.(L3)
- **CO-3.** Compare different measures of computer performance and calculate relevant performance metrics to assess system efficiency.(L4)
- **CO-4.** Analyze the organization and structure of processors, including register organization, instruction cycle, and the concept of instruction pipelining.(L4)
- **CO-5.** Evaluate the design issues related to instruction-level parallelism and superscalar processors. (L5)

3. Course Contents

Sr. No	Units
1	**Introduction to the computer system**: **Basic Concepts and Computer Evolution**: Organization and Architecture, Structure and Function, A Brief History of Computers, The Evolution of the Intel x86 Architecture, Embedded Systems **Performance Issues**:Two Laws that Provide Insight: Ahmdahl's Law and Little's Law, Basic Measures of Computer Performance, Calculating the Mean, Benchmarks and Spec. **Top-Level View of Computer Function and Interconnection**: Computer Components, Computer Function, Interconnection Structures, Bus Interconnection, Point-to-Point Interconnect **Memory**: Computer Memory System Overview, Cache Memory Principles, Elements of Cache Design, Semiconductor Main Memory, DDR DRAM **Input/Output**: External Devices, I/O Modules, Programmed I/O, Interrupt-Driven I/O, Direct Memory Access

2	**The Central Processing Unit** **Instruction Sets:** Characteristics and Functions:Machine Instruction Characteristics, Types of Operands, Types of Operations **Instruction Sets:** Addressing Modes and Formats: Addressing Modes, Instruction Formats, Assembly Language
3	**The Processor** **Processor Structure and Function:**Processor Organization, Register Organization, Instruction Cycle, Instruction Pipelining **Instruction-Level Parallelism and Superscalar Processors:**Overview, Design Issues, Intel Core Microarchitecture
4	**Parallel Organization** **Parallel Processing:**Multiple Processor Organizations, Symmetric Multiprocessors, Cache Coherence and the MESI Protocol, Multithreading and Chip Multiprocessors **Multicore Computers:**Hardware Performance Issues, Software Performance Issues, Multicore Organization, Heterogeneous Multicore Organization.
5	**General-Purpose Graphic Processing Units** Cuda Basics, GPU versus CPU, GPU Architecture Overview **Control Unit Operation** Micro-Operations , Control of the Processor , Hardwired Implementation, Microprogrammed control.

4. Course Articulation Matrix (CO-PO-PSO Map)

	Programme Outcomes (POs)												**Programme Specific Outcomes (PSOs)**			
	PO-1	**PO-2**	**PO-3**	**PO-4**	**PO-5**	**PO-6**	**PO-7**	**PO-8**	**PO-9**	**PO-10**	**PO-11**	**PO-12**	**PSO-1**	**PSO-2**	**PSO-3**	**PSO-4**
CO-1	3										2		3		1	1
CO-2	3										2		3		2	1
CO-3	3										2		3		2	1
CO-4	3	3									2		3	2	2	2
CO-5	3	3	2	2							2		3	2	2	2
	3: Very Strong Contribution, 2: Strong Contribution, 1: Moderate Contribution															

5. Course Resources

a. Essential Reading

1. Course Self-Learning Material
2. William Stallings, Computer Organization and Architecture Designing for Performance, 10th Ed, Pearson Education, 2016.

b. Recommended Reading

1. John L. Hennessy and David A. Patterson, Computer Architecture: A Quantitative Approach 5th Edition, Elsevier publication, 2017.
2. Patterson, J.L. Hennessy, Computer Organization and Design: The Hardware/Software Interface, Morgan Kaufmann, 5th edition, 2013

c. Websites

1. https://www.coursera.org/
2. http://nptel.ac.in/

OMCxxx – Green Computing

Program	Master of Computer Applications
Semester	1
Course Title	Green Computing
Course Code	OMCxxx
Course Credits	3
Course Type	Core Course

1. Course Summary

The Green Computing course is designed to introduce students to the principles, practices, and technologies involved in sustainable computing. The course focuses on the environmental impact of information and communication technologies (ICT) and explores methods to reduce energy consumption, promote resource efficiency, and minimize electronic waste. Students will gain a comprehensive understanding of the concepts and strategies behind green computing and its relevance in today's digital world.

2. Course Outcomes (COs)

After the successful completion of this course, the student will be able to:

- **CO-1.** Understand the concept of Green Computing and its significance in promoting environmental sustainability within the IT industry. [L-2].
- **CO-2.** Apply energy efficiency techniques to computing systems, including power management [L-3].
- **CO-3.** Evaluate the potential of renewable energy sources and their application in computing infrastructure[L-5].
- **CO-4.** analyze resource management strategies and waste reduction practices in computing environments[L-4].
- **CO-5.** Demonstrate knowledge of virtualization, cloud computing, and data centers in achieving green IT objectives. [L-5].
- **CO-6.** Apply sustainable software development practices, including energy-aware programming and software optimization. [L-4].

3. Course Contents

Sr. No	Units	Unit Outcomes After the successful completion of the unit, the learner should be able to:
1	**Unit 1: Introduction to Green Computing** • Definition, goals, and benefits of Green Computing • Environmental challenges in computing • Importance of sustainability in the IT industry • Power management techniques	1. Understand the concept of Green Computing and its importance in addressing environmental challenges in the IT industry. 2. Identify the goals and benefits of Green Computing practices. 3. Analyze the environmental impact of conventional computing practices and the need for sustainable alternatives.

2	**Unit 2: Resource Management, Waste Reduction and Renewable Energy** • Virtualization and server consolidation • Efficient cooling and data center design • E-waste management and recycling • Green procurement and disposal practices • Solar, wind, and other renewable energy sources • Renewable energy applications in data centers • Energy harvesting techniques • Green energy certifications and standards	1. Implement virtualization and server consolidation techniques to optimize resource utilization. 2. Develop strategies for e-waste management and implement recycling practices in computing environments. 3. Assess the potential of renewable energy sources and their integration into computing infrastructure.
3	**Unit 3: Cloud Computing and Green Data Centers** • Virtualization and cloud computing technologies • Green data center design and operation • Energy-efficient storage and networking • Green cloud service providers and sustainability considerations. •	1. Analyze the role of virtualization and cloud computing in achieving green IT objectives. 2. Design and implement energy-efficient storage and networking solutions in cloud environments. 3. Evaluate and select green cloud service providers based on sustainability considerations.
4	**Unit 4: Sustainable Software Development** • Green software engineering practices • Energy-aware programming techniques • Software optimization and performance tuning • Green metrics and environmental impact assessment	1. Apply green software engineering practices to develop energy-efficient software solutions. 2. Optimize software performance and resource utilization for reduced energy consumption. 3. Assess the environmental impact of software development decisions and propose sustainable alternatives.
5	• **Unit 5: Environmental Considerations in IT Infrastructure** • Life cycle assessment and eco-design • Green IT policy and governance • Sustainable IT procurement and supply chain management • Green data management and storage solutions	1. Conduct life cycle assessments and implement eco-design principles in IT infrastructure planning. 2. Develop and implement green IT policies and governance frameworks. 3. Evaluate and implement sustainable IT procurement and supply chain management practices.

4. Course Articulation Matrix (CO-PO-PSO Map)

	Programme Outcomes (POs)												**Programme Specific Outcomes (PSOs)**			
	PO-1	**PO-2**	**PO-3**	**PO-4**	**PO-5**	**PO-6**	**PO-7**	**PO-8**	**PO-9**	**PO-10**	**PO-11**	**PO-12**	**PSO-1**	**PSO-2**	**PSO-3**	**PSO-4**
CO-1				1	2	2	3					2	1		3	
CO-2		2	2				3	2				2		3		2
CO-3				2		2	2					2		3	2	
CO-4				3	2		3					1	1		3	
CO-5		2	2			3	2	3				3		2		2
CO-6				3	2		1	1				2			3	2
		3: Very Strong Contribution, 2: Strong Contribution, 1: Moderate Contribution														

5. Course Resources

a. Essential Reading

1. Bud E. Smith, 2014, Green Computing: Tools and Techniques for Saving Energy, Money, and Resources Press, Taylor & Francis Group.
2. Toby Velte, Anthony Velte, and Robert Elsenpeter, 2008, Green IT: Reduce Your Information System's Environmental Impact While Adding to the Bottom Line. Velte Publishing.
3. Corey Gough, Ian Steiner, and Winston Saunders, 2015, Energy Efficient Servers: Blueprints for Data Center Optimization, Apress open Publishing.

b. Recommended Reading

1. Supriya Kumar De and Satchidananda Dehuri, Green Computing: Advanced Topics and Case Studies.
2. Pierre Bonnet, Jean-Michel Detavermier, Dominique Vauquier, 2009, Sustainable IT Architecture: The Progressive Way of Overhauling Information Systems with SOA. Wiley Publications.
3. Hideharu Amano, Tadahiro Kuroda, and Tomoya Fujii, 2013, Green Computing with Emerging Memory: Low-Power Computation for Social Innovation, Springer Publications.

c. Magazines and Journals

a. International Journal of Green Computing(IGI Global Publications).
b. Journal of Smart Environments and Green Computing.

d. Websites

1. https://www.coursera.org/
2. http://nptel.ac.in/

e. Other Electronic Resources

1. Course Video Lectures on Brightspace

MCA105 - Discrete Structures and Combinatorics

Program	Master of Computer Applications
Semester	1
Course Title	Discrete Structures and Combinatorics
Course Code	MCA 105
Course Credits	3
Course Type	Core Theory Course

1. Course Summary

The goal of this course is to lay a strong foundation for the discrete structures and combinatorics among learners. This course has been started by explaining the basics of set theory, relations and functions for a better understanding of the learners. The learners are taught the importance of preposition logic and mathematical induction. The learners are also taught how to compare and analyze mathematical problems in the context of group theory. In the subsequent lessons, various types of graph theory related concepts have been discussed and explained to the learners. In this course, the concepts of combinatorics have been discussed in depth including Fundamental Principles, Factorial Notations, Permutations, Combinations, Binomial Theorem, and Multimonial coefficients. In this course, various concepts related to counting principles, such as Pigeonhole Principle and Inclusion-Exclusion Principle have been explained to the learners. This course also emphasizes Discrete Numeric Functions including Recurrence relations and Generating Functions.

2. Course Outcomes (COs)

After the successful completion of this course, the student will be able to:

CO-1. Define basic mathematical objects such as sets, functions, relations and other mathematical structures [L-1]
CO-2. Express mathematical properties formally via the formal language of propositional logic [L-2]
CO-3. Perform various techniques of mathematical induction (weak, strong and structural induction) on variety of discrete structures [L-3]
CO-4. Categorize and analyze problems based on group theory and graph theory [L-4]
CO-5. Determine and select appropriate counting techniques to solve combinatorial problems [L-5]

3. Course Contents

Sr. No	Units
1	**Unit 1: Sets, Relations, and Functions** **Sets** • Introduction • Countable and Uncountable Set • Sets and Subsets • Basic Operations on Sets • Algebraic properties of set operations **Relations:** • Introduction • Cartesian Product • Relations and their types • Properties of Relations • Equivalence Relations • Partial order relations • Hasse's diagram **Functions:** • Introduction • Composition of functions • Inverse of functions • Recursively defined functions

	• Functions for Computer Science ➢ Floor and ceiling Function ➢ Ackermann's Function ➢ Mod Function
2	**Unit 2: Propositional Logic and Mathematical Induction** **Propositional Logic** • Introduction • Basic logical operations • Tautologies • Contradictions • Algebra of proposition • Logical implication • Logical equivalence and Validity **Mathematical Induction** • Introduction of Mathematical Induction • Normal forms • Rules of Inference • Predicates and Quantifiers
3	**Unit 3: Group Theory:** • Introduction • Semi group • Monoid • Group • Abelian group • Subgroup and their properties • Cyclic group • Cosets • Lagrange's theorem • Permutation groups **More About Groups:** • Introduction • Homomorphism • Isomorphism • Automorphism of groups

4	**Unit 4: Graph Theory** • Definition of graphs • Applications of graph • Types of graphs • Subgraph • Isomorphic graph • Eulerian graph • Hamiltonian graph **Operations on Graphs:** • Operations on graphs • Representation of graphs • Planar graphs • Coloring of graphs
5	**Unit 5: Combinatorics and Discrete Numeric Functions** **Combinatorics:** • Introduction • Fundamental Principles • Factorial Notations • Permutations • Combinations • Binomial Theorem • Multimonial coefficients **Some More Counting Principles:** • Pigeonhole Principle • Inclusion-Exclusion Principle • Applications of Inclusion – Exclusion ➢ Application to Surjective Functions ➢ Application to Derangements **Discrete Numeric Functions:** • Introduction • Discrete Numeric Functions • Recurrence relations • Generating Functions

4. Course Articulation Matrix (CO-PO-PSO Map)

	Programme Outcomes (POs)												Programme Specific Outcomes (PSOs)			
	PO-1	**PO-2**	**PO-3**	**PO-4**	**PO-5**	**PO-6**	**PO-7**	**PO-8**	**PO-9**	**PO-10**	**PO-11**	**PO-12**	**PSO-1**	**PSO-2**	**PSO-3**	**PSO-4**
CO-1	3	3	3	3							2	2		3	1	
CO-2	3	3	3	3							2	2	3	3	1	1
CO-3	3	3	3	3							2	1		2	1	1
CO-4	3	3	3	3	3						3	2	3	3	3	2
CO-5	3	3	3	3	2						3	2	2	2		1
		3: Very Strong Contribution, 2: Strong Contribution, 1: Moderate Contribution														

5. Course Resources

a. Essential Reading

1. Course Self-Learning Material
2. Sengadir, T., 2009, Discrete Math And Combinatorics, Pearson Education.

b. Recommended Reading

1. Lipschutz, S. and Lipson, M., 2009, Discrete Mathematics, Tata McGraw-Hill.
2. Tremblay, J. P. and Manohar, R., 1997, Discrete Mathematical Structure with Application to Computer Science, Tata McGraw-Hill.
3. Alan, D. and Kenneth, L., 2000, Applied Discrete Structures for Computer Science, Galgotia Pub. Pvt. Ltd.

c. Magazines and Journals

1. SIAM Journal on Discrete Mathematics
2. IEEE Transaction on Information Theory
3. Journal of Combinatorial Theory Series A, Academic Press
4. Graphs and Combinatorics, Springer

d. Websites

1. https://egyankosh.ac.in/handle/123456789/951
2. https://www.coursera.org/specializations/discrete-mathematics
3. https://nptel.ac.in/courses/106108227/
4. https://nptel.ac.in/courses/111106155

e. Other Electronic Resources

1. https://ocw.mit.edu/index.htm
2. Course Video Lectures on ILearn

--**--

OMCxxx – Cloud Computing

Program	Master of Computer Applications
Semester	1
Course Title	Cloud Computing
Course Code	OMCxxx
Course Credits	3
Course Type	Core Course

1. Course Summary

The course provides a comprehensive overview of cloud computing, covering key topics such as an introduction to the cloud, compute services, storage services, database services, and network services. Students gain a solid understanding of the fundamental concepts and principles of cloud computing, including its significance in modern technology. They explore different types of compute services, such as virtual machines and containers, and learn about configuring and managing compute nodes. The course also delves into various storage options available in the cloud, including file storage, block storage, and storage for backups, while discussing their advantages and considerations. Students gain insights into cloud-based database services, including relational databases, key-value databases, and time series databases, and understand the architectural considerations and data models associated with databases in the cloud. Lastly, they explore network services in the cloud, focusing on building and scaling cloud networks, securing network traffic, and the role of content delivery networks.

2. Course Outcomes (COs)

After the successful completion of this course, the student will be able to:

CO-1. Classify various cloud computing services and models[L-2].
CO-2. Use different compute services in cloud with a case study[L-3].
CO-3. Analyze the benefits and challenges of using cloud-based data storage in comparison to traditional on-premises storage. [L-4].
CO-4. Evaluate the trade-offs between different database features and characteristics, such as consistency, durability, and query capabilities. [L-5].
CO-5. Illustrate various security mechanisms and services available for securing network traffic, such as virtual private clouds (VPCs), network security groups (NSGs), and web application firewalls (WAFs). [L-4].
CO-6. Explain the concept and benefits of a content delivery network (CDN) in distributing and delivering content to users. [L-2].

3. Course Contents

Sr. No	Units	Unit Outcomes After the successful completion of the unit, the learner should be able to:
1	**Unit 1: Fundamentals of Cloud Computing** • Introduction • Cloud Types • Deployment Models • Service Models • Virtualization • Cloud architecture • **Case Study: Amazon Web Services, Microsoft Azure.**	4. Discuss the evolution of cloud computing and its impact on various industries. 5. Analyze real-world examples of organizations using different cloud types and their use cases. 6. Describe the components and layers of cloud architecture (e.g., infrastructure, platform, software).
2	**Unit 2: Compute Services of Cloud Computing** • Compute Node Architecture • Types of Compute nodes: Virtual Machines & Containers • Configuration of Compute node. • **Case study: Amazon EC2, Virtual Box, Docker.**	4. Explain the components and functionalities of a compute node, such as processors, memory, storage, and networking. 5. Discuss the advantages and disadvantages of using VMs and containers for deploying applications. 6. Describe the process of configuring a compute node for deployment.
3	**Unit 3: Storage Services in Cloud Computing** • Introduction to Cloud based Data Storage • Advantages and disadvantages of Cloud based data Storage • Types of cloud storage: File storage, • Block storage-Elastic Block Storage. • Storage for backups • **Case Study: Amazon S3, EBS, EFS, Glacier**	4. Analyze the benefits and challenges of using cloud-based data storage in comparison to traditional on-premises storage. 5. Identify the advantages of cloud-based data storage, such as scalability, cost-effectiveness, and accessibility. 6. Describe block storage and its use in cloud environments, specifically focusing on Elastic Block Storage (EBS) offered by cloud providers.
4	**Unit 4: Database Servies in Cloud Computing** • Need for Cloud Databases • Consideration for databases	4. Explain the reasons and benefits of using cloud databases in modern data management. 5. Discuss the factors to consider when selecting a suitable database solution for specific use cases, including workload requirements, data

	• Database architecture • Data Models • Relational Databases • key-value based databases • Time series databases • **Case study: Amazon RDS, DynamoDB**	volume, and data access patterns. 6. Explain the components and layers of a typical database architecture, including the storage layer, query processing layer, and access interfaces.
5	**Unit 5: Networking and Security Services** • Building Cloud network • Scaling Cloud Network • Securing Network Traffic • Content Delivery Network • **Case study: Amazon VPC, Route S3**	1. Discuss the best practices for configuring networking services, such as load balancers, virtual private networks (VPNs), and firewalls, in a cloud environment. 2. Explain the concepts of horizontal and vertical scaling in the context of cloud networks. 3. Identify the security risks and threats associated with network traffic in a cloud environment. 4. Discuss the architecture and components of a CDN, including edge servers, caching mechanisms.

4. Course Articulation Matrix (CO-PO-PSO Map)

	Programme Outcomes (POs)												**Programme Specific Outcomes (PSOs)**			
	PO-1	**PO-2**	**PO-3**	**PO-4**	**PO-5**	**PO-6**	**PO-7**	**PO-8**	**PO-9**	**PO-10**	**PO-11**	**PO-12**	**PSO-1**	**PSO-2**	**PSO-3**	**PSO-4**
CO-1	3	2	1									2	3		2	
CO-2	2	3	3	2	3							3	2	3		2
CO-3	1	2	3	2	3							3	2		3	
CO-4		2	2	3	3							2	2	3		
CO-5		2	3	3	3							3		2		3
CO-6		3	2	3	2							2			2	3
	3: Very Strong Contribution, 2: Strong Contribution, 1: Moderate Contribution															

5. Course Resources

a. Essential Reading

1. Furht, Borivoje, and Armando Escalante, 2010, Handbook of cloud computing. Vol. 3. New York: springer.
2. Kavis, Michael J., 2014, Architecting the cloud: design decisions for cloud computing service models (SaaS, PaaS, and IaaS). John Wiley Sons.
3. Vacca, John R., ed. 2016, Cloud computing security: foundations and challenges. CRC Press.

b. Recommended Reading

1. Wittig, Michael, Andreas Wittig, and Ben Whaley, 2018, Amazon web services in action. Manning.
2. Kailash Jayaswal, Jagannath Kallakurchi, Donald J. Houde, Deven Shah,2014, Cloud Computing Black Book, Dreamtech press

c. Magazines and Journals

a. IEEE Transaction on Cloud Computing.
b. Journal of Cloud Computing.

d. Websites

1. https://www.coursera.org/
2. http://nptel.ac.in/

e. Other Electronic Resources

1. Course Video Lectures on Brightspace

MCA 1st Semester – Full Stack Development Laboratory

Program	Master of Computer Applications
Semester	1
Course Title	Full Stack Development Laboratory
Course Code	OMC xxx
Course Credits	2
Course Type	Laboratory

1. Course Summary

The aim of this course is to gain the skills and knowledge necessary to build simple web applications as well as full-stack web applications using modern and scalable web technologies and increase employability as a full-stack developer. The students are taught the basics of HTML, CSS, JavaScript, PHP, and the basic components of Full Stack development using MERN stack widely used in the industry for developing web pages. Students will learn the use of XHTML and CSS for developing presentable web pages. They will also be able to create dynamic web pages by applying event-handling mechanisms using JavaScript. Students will understand the concepts of cookies and sessions in PHP for creating large web applications. Students will learn to develop simple ReactJS applications.

2. Course Outcomes (COs)

After the successful completion of this course, the student will be able to:

CO-1. Demonstrate the usage of XHTML tags, and CSS for developing presentable web pages.[L-3]

CO-2. Develop dynamic web pages by applying event-handling mechanisms using JavaScript.[L-5]

CO-3. Demonstrate the use of COOKIES and SESSION in PHP. [L-3]

CO-4. Develop simple ReactJS applications.[L-5]

3. Course Contents - Laboratory Programs List

1. Create an XHTML page that provides information about the MCA department, at Graphic Era University. The XHTML page must use the following tags:
    - Anchor tag
    - Images
    - Links
    - Tables

    (if needed use other tags for better presentation)
2. Create an XHTML page that demonstrates the usage of lists and tables.
3. Create an XHTML page that displays a Form with all types of controls (Text Boxes, Radio buttons, Checkboxes, Dropdown, Submit and Reset buttons) on it with proper formatting.
4. Develop a web page and demonstrate the usage of inline style, internal style, and external style sheets using CSS.
5. Write a JavaScript function called "MaxandMinofArray" that accepts an array of integers as a parameter and displays the largest and smallest number in the array. Test the function with different inputs. Embed the JavaScript function within the XHTML document.
6. Write a JavaScript function called "SumofDigits" that accepts a number as a parameter and returns the sum of all digits of that number. Test the function with different inputs. Write the JavaScript function in a separate .js file.
7. Create an XHTML document with two buttons. Write a JavaScript function that triggers an alert message when the button is clicked. It should display the message "First button is clicked" or “Second button is clicked" depending on the button being clicked.
8. Create an XHMTL page with 3 paragraphs displayed using different colors. Implement a JavaScript function that changes the font color of a paragraph to blue when a user hovers over it and reverts it back to the original color when the mouse leaves.
9. Create an XHTML document with a form that collects the mobile number. On submitting the form validate the input using an event handler. The mobile number should be a 10-digit number. On validating display, a success or failure message using "alert()".
10. Write a PHP program using COOKIE to store the current date and time and on reopening the same web page display the "Last visited date and time".
11. Write a PHP program to demonstrate the use of SESSIONS to increment a count on each page refresh, and display the same on the web page.
12. Create a React Application to display the message “Developing using ReactJS, Graphic Era University”.

4. Course Articulation Matrix (CO-PO-PSO Map)

	Programme Outcomes (POs)												Programme Specific Outcomes (PSOs)			
	PO-1	**PO-2**	**PO-3**	**PO-4**	**PO-5**	**PO-6**	**PO-7**	**PO-8**	**PO-9**	**PO-10**	**PO-11**	**PO-12**	**PSO-1**	**PSO-2**	**PSO-3**	**PSO-4**
CO-1	3	2	3		3			2		1	2		3	3		
CO-2	3	2	3		3			2		1			3	3		
CO-3	3	3	3		3	2		2		1		2	3	3	3	2
CO-4	3	3	3		3	2		2	2	1		2	3	3	3	3
	3: Very Strong Contribution, 2: Strong Contribution, 1: Moderate Contribution															

5. Course Resources

a. Essential Reading

1. Robert W. Sebesta, "Programming the world wide web", 6th edition, Pearson education.
2. Vasan Subramanian, "Pro MERN Stack, Full Stack Web App Development with Mongo, Express, React and Node", 2nd Edition, Apress.

b. Recommended Reading

1. Kogent Learning Solutions Inc., "HTML 5: Covers CSS3, JavaScript, XML, XHTML, AJAX, PHP & jQuery: Black Book", Dreamtech Press.
2. Eddy Wilson, 2018, "MERN Quick start guide: Build Web applications with MongoDB, Express.js, React and Node", Packt publishing.

c. Websites

1. https://www.coursera.org/
2. http://nptel.ac.in/

--**--

OMCXXX – Operating system & Computer Networks Laboratory

Program	Master of Computer Applications
Semester	1
Course Title	Operating system & Computer Networks Laboratory
Course Code	OMC102
Course Credits	2
Course Type	Core Course

1. Course Summary

The Operating System and Computer Network Lab course offers practical experience in computer networks and operating systems. Students learn network commands, configure network

devices, implement point-to-point networks, and analyze network performance. They also simulate CPU scheduling algorithms, memory management techniques, disk scheduling algorithms. This hands-on approach enhances their understanding of network protocols, resource allocation, and system performance. By completing the term works, students gain valuable skills in network configuration, command-line tools, and practical implementation of operating system algorithms..

2. Course Outcomes (COs)

After the successful completion of this course, the student will be able to:

- **CO-1.** Explain the use of network commands in command prompt to diagnose and troubleshoot network connectivity issues. L2)
- **CO-2.** Configure and demonstrate the use of hub, switch, and router in a simple network topology.(L3)
- **CO-3.** Analyze the impact of queue size and bandwidth on network performance by implementing a point-to-point network with varying parameters and observing packet drops.(L4)
- **CO-4.** Simulate CPU scheduling algorithms such as FCFS and SJF and analyze their performance using different process arrival times and burst times .(L4)
- **CO-5.** Implement an Ethernet LAN with multiple traffic nodes, and evaluate congestion window behavior for different source/destination pairs. (L5)

3. Course Contents

Sr. No	**Units**
	Computer Networks
1	Demonstrate network commands in command promt.(like ping,traceroute)
2	Demonstrate use of hub, switch, router using simple topology.
3	Implement three nodes point – to – point network with duplex links between them. Set the queue size, vary the bandwidth and find the number of packets dropped.
4	Implement an Ethernet LAN using n nodes and set multiple traffic nodes and plot congestion window for different source / destination.
5	Demonstrate static routing.
6	Demonstrate dynamic routing protocols.

	Operating systems
7	Simulate the following CPU scheduling algorithms-FCFS,SJF
8	Simulate the following Memory management algorithm-First fit, Best fit
9	Implement the optimal page replacement algorithm
10	Implement the FCFS Disk scheduling and SSTF Disk scheduling
11	Implement the producer consumer problem with solution using semaphore
12	Implement a program in C to extract process ID (PID) and parent process ID (PPID).

Note: The computer networks termworks can be implemented using a suitable simulator like CISCO packet tracer, GNS3 etc. The opertating system termworks can be implemented using a suitable programming language like C,C++ etc.

4. Course Articulation Matrix (CO-PO-PSO Map)

	Programme Outcomes (POs)												**Programme Specific Outcomes (PSOs)**			
	PO-1	**PO-2**	**PO-3**	**PO-4**	**PO-5**	**PO-6**	**PO-7**	**PO-8**	**PO-9**	**PO-10**	**PO-11**	**PO-12**	**PSO-1**	**PSO-2**	**PSO-3**	**PSO-4**
CO-1	3										2		2	2	2	2
CO-2	3										2		2	2	2	2
CO-3	3	2									2		2	2	2	2
CO-4	3	2		2							2		2	2	2	2
CO-5	3	2	2	2	2						2		2	2	2	2
	3: Very Strong Contribution, 2: Strong Contribution, 1: Moderate Contribution															

5. Course Resources

a. Essential Reading

1. Course Self-Learning Material

b. Recommended Reading

1. James F. Kurose and Keith W. Ross, 2017, Computer Networking: A Top-Down Approach, 7th Edition,Prentice Hall.
2. Abraham Silberschatz, Peter B Galvin, Greg Gagne, “Operating System Concepts”, Wiley India Pvt. Ltd 2018, 9th Edition

c. Websites

1. https://www.coursera.org/
2. http://nptel.ac.in/

MCA 1st Semester – Programming and Problem-Solving Laboratory

Program	Master of Computer Applications
Semester	1
Course Title	Programming and Problem-Solving Laboratory
Course Code	OMC XXX
Course Credits	2
Course Type	Laboratory

1. Course Summary

The aim of this course is to gain the skills, hands-on experience, and practical knowledge necessary to write programs using 'C language. The Students will learn to develop algorithms, draw flowcharts and write a 'C' program to solve a given problem. They will learn to use the online GDB 'C' Debugger/Compiler for programming, debugging, and executing the programs. The students are taught the basics of 'C' Programming which also provides a base for learning any other computer-related languages. Students will learn the use of conditional statements and loops, explore the concepts of arrays, and understand the concepts of modular programming and reusable code using functions. Students learn about memory management and dynamic data structures through the use of pointers. They will also be able to create user-defined data types called structures. Additionally, they will understand file-handling mechanisms for storing and retrieving data. Students will learn the techniques of debugging and testing 'C' programs.

2. Course Outcomes (COs)

After the successful completion of this course, the student will be able to:

CO-1. Develop an algorithm, draw a flowchart, and write a 'C' program to solve a given problem. [L-3]

CO-2. Make use of online GDB 'C' Debugger/Compiler for programming, debugging, and executing the programs.

CO-3. Demonstrate the use of expressions, decision structures, loops, functions, library functions, recursive functions, arrays, strings, structures, pointers, and files in problem-solving.[L-3]

CO-4. Document the conclusion and observations made from the implementation.[L-4]

3. Course Contents - 'C' Programming Laboratory Program List

1. Write a 'C' program to evaluate an expression $ax^2 + bx + c$.
2. Write a 'C' program to find the sum of the first n even natural numbers by using the formula - n(n+1)
3. Write a 'C' program to find whether the given year is a Leap year.
4. Write a 'C' program to accept input and print whether the given input is a character, digit, or special symbol using a switch-case.
5. Write a 'C' program to accept the month number and print the month name using switch-case
6. Write a 'C' program to find the maximum and minimum numbers of an array of integers.
7. Write a 'C' program to find the transpose of a given matrix.
8. Write a 'C' program that finds the sum of diagonal elements of a given square matrix.
9. Write a 'C' program to read an integer and find the sum of digits. Use a function for finding the sum of digits.
10. Write a 'C' program to find convert a decimal number to a binary number. Use recursive function.
11. Write a C program to perform the following operations on a string.
    a) Count the number of characters and digits.
    b) Check whether the given string is palindrome or not.
12. Write a 'C' program to swap two numbers. [Demonstrate the difference between call by value and call by reference for swapping the two numbers]
13. Write a 'C' program to store and retrieve the personal information about students – Enrolment Number, Name, Address, City, and State. Demonstrate using files.
14. Write a 'C' program to find
    a) The frequency of characters that are in a sentence stored in a text file.
    b) Count the number of words in a file.

    The file name is to be supplied as a command-line argument.

4. Course Articulation Matrix (CO-PO-PSO Map)

	Programme Outcomes (POs)												**Programme Specific Outcomes (PSOs)**			
	PO-1	**PO-2**	**PO-3**	**PO-4**	**PO-5**	**PO-6**	**PO-7**	**PO-8**	**PO-9**	**PO-10**	**PO-11**	**PO-12**	**PSO-1**	**PSO-2**	**PSO-3**	**PSO-4**
CO-1	3	2	3		3			2		1	2		3	3		1
CO-2	3	2	3		3			2		1			3	3		1
CO-3	3	3	3		3	2		2		1	2	2	3	3	3	2
CO-4	3	3	3		3	2		2	2	1	2	2	3	3	3	3
	3: Very Strong Contribution, 2: Strong Contribution, 1: Moderate Contribution															

5. Course Resources

a. Essential Reading

1. David D. Riley and Kenny A. Hunt, (2014), "Computational thinking for the Modern Problem Solver", Chapman & Hall/CRC.

2. Yashavant Kanetkar, (2016), “Let Us C”,14th Edition, BPB Publication.

b. Recommended Reading

1. E. Balagurusamy, (2015), “Programming in ANSI C”,6th Edition, McGraw-Hill.
2. Brian W Kernighan & Dennis M Ritchie, (1988), “The C Programming Language”, 2nd Edition, Prentice Hall.
3. Steve Oualline,(2011), “Practical C Programming”, 3rd Edition, Orielly Publishers.

c. Websites

5. https://www.coursera.org/
6. http://nptel.ac.in/

--**--

Professional Communication and Career Skills-I

Learning Outcomes:

LO 1 Apply logic to the different types of arrangement-based questions and arrive at solutions to deterministic and nondeterministic question sets.

LO 2 Apply the concept of blood relations and learn to draw a family tree using the different notations.

LO 3 Solve the different types of questions based on orientation of direction and understanding of distances and turns.

LO 4 Learn to construct a Venn diagram using multiple statements and arrive at conclusions or possibilities based on logic.

LO 5 Comprehend different types of data sets used in Data Interpretation and use quick calculation techniques for solving different types of questions.

UNIT 1: **3 Hours**

Introduction to reasoning, basic concepts, and practice of deterministic and nondeterministic arrangement-based questions (linear, vertical, circular and rectangular).

Concepts and understanding of deterministic and nondeterministic tabular or grid-based questions including understanding of variables and their entries in the solution table. Practice of tabular or grid-based question sets.

UNIT 2: **3 Hours**

Blood relation concepts include basic introduction, making a family tree, standard notations and names for gender and relations. Discussion of different types of questions asked in blood relations, their solutions and practice.

Concepts and practice of grouping/team formation or condition-based questions including the understanding and application of different conditions used in grouping sets.

UNIT 3: **3 Hours**

Basic concept and understanding of directions including the orientation of the 4 basic directions of east, west, north, and south. Understanding turns of different degrees towards right, left, clockwise and anticlockwise.

Basic concept of coding-decoding using alphabets, digits, words and their combinations. Understanding and practice of different questions in coding decoding.

Basic concept of series completion using numbers, alphabets, and their combinations thereof. Understanding of different types of series (based on differences, based on products, based on exponentials).

UNIT 4: **3 Hours**

Introduction to Data Interpretation (DI), understanding different methods of data representation including tabular, bar graph, pie chart, line graph and caselet. Techniques of quick arithmetic calculations, concepts of percentage as applicable in DI, growth and growth rate and practice of various DI sets.

UNIT 5: **3 Hours**

Understanding the concepts of Syllogism using Venn diagram, types of problems in syllogism (2 statements, 3 statements and 6 statement problems).

Reference books and study material:

1. Lalit Singh and P.A.Anand, Verbal Ability and Reasoning for Competitive Exams, Wiley, First Edition, January 2016.

2. R.S.Aggarwal, A Modern approach to Verbal and Non-Verbal Reasoning for Competitive Exams, S Chand Publication, January 2018

3. Shakuntala Devi, Puzzles to Puzzle you, Orient Paperbacks, June 1976.

4. George Summers, The great book of Puzzles and Teasers, Jaico Publishing, September 1989.

5. P.A.Anand, reasoning book, Savera publication.

6. Any other reading as suggested by the faculty.