Chaudhary Ranbir Singh University, Jind

(Established vide Haryana State Legislative Act 28 of 2014)
(Recognised u/s 2(f) and 12(B) of UGC Act, 1956)

Scheme of Examination
for
Post Graduate Programme

Master of Computer Applications (MCA)

as per NEP 2020
Curriculum and Credit Framework for Postgraduate Programme

with Multiple Entry-Exit, Internship and CBCS-LOCF
with effect from the session 2024-25 (in phased manner)

DEPARTMENT OF COMPUTER SCIENCE & APPLICATIONS
FACULTY OF PHYSICAL SCIENCES

CHAUDHARY RANBIR SINGH UNIVERSITY
JIND – HARYANA – INDIA - 126102

Vishal

Abbreviations used

Sr. No.	Full form	Abbreviation	Description
1	Core Course	CC	Compulsory core courses for the programme. CC will be a theory course of 4 credits.
2	Discipline Elective Course	DEC	Elective Courses offered by the DCI. A student can opt one course out of 4 given options for that DEC course. One course can be opted in a semester through MOOCs from SWAYAM or other portals. DEC will be a theory course of 4 credits.
3	Practicum	PC	Practical course of 4 credits which will be compulsory in all semesters for all students except in the 4th Semester when a student opts Dissertation work.
4	Seminar	S	The seminar is a Skill Enhancement Course (SEC) aiming to impart skills of self-learning, comprehension, communication and presentation.
5	Constitutional, Human, Moral Values and IPR	CHM	CHM is a compulsory Value Added theory Course of 2 credits.
6	Open Elective Course	OEC	OEC is a Multidisciplinary course of 2 credits. Every student will opt for a course from the pool of OEC courses other than Computer Science.
7	Employability and Entrepreneurship Skills Course	EEC	EEC is a Vocational or SEC course aiming to increase the employment and entrepreneurship potential of students of programme.
8	Theory	Th	
9	Practical	P	
10	Lecture	L	
11	Tutorial	T	
12	Dissertation	D	A research course of 12 credits, where a student will undertake research work and submit a dissertation as per rules prescribed by the university.
13	Programme Learning Outcomes	PLOs	
14	Course Learning Outcomes	CLOs	

Vitthal

Programme Learning Outcomes(PLOs): As per NEP-2020, PLOs include outcomes specific to disciplinary areas of learning associated with the chosen field (s) of learning as well as generic learning outcomes. These also include transferable skills and competencies that post-graduates of all programmes of study should acquire and be able to demonstrate for the award of the Degree. The programme learning outcomes would also focus on knowledge and skills that prepare students for further study, employment, research, and responsible citizenship.

The PLOs of the **MCA** programme arestated as per the following domains:

PLOs	After the completion of MCA degree, a student will be able to:
PLO-1: Knowledge and Understanding	Demonstrate the deep understanding and advanced knowledge in the core areas of Computer Science subject and understanding of recent developments and issues, including concepts, theories, principles, methods, and techniques in different areas of Computer Science.
PLO-2: General Skills	Acquire the general skills required for performing and accomplishing the tasks as expected to be done by a skilled professional in the fields of Computer Applications.
PLO-3: Technical/ Professional Skills	Demonstrate the learning of advanced cognitive computing, programming, formulating models, using various softwares, and other teaching and professional skills required for completing the specialized tasks related to the profession and for conducting and analyzing the relevant research tasks in different domains of Computer Applications.
PLO-4: Communication Skills	Effectively communicate the attained skills in different areas of Computer Sciencein a precise, well-structured, and unambiguous mathematical language through effective oral and/or written expressions to the society at large.
PLO-5: Application of Knowledge and Skills	Apply the acquired knowledge and skills to the problems in the subject area, and identify and analyze the issues where the attained knowledge and skills can be applied by carrying out various industry-oriented projects and/or research investigations to formulate appropriate solutions to various problems ranging from basic to complex and unpredictable problems associated with the field of Computer Applicationsor allied fields.
PLO-6: Critical Thinking and Research Aptitude	Attain the capabilities of critical thinking, logical reasoning, investigating problems, analysis, problem-solving, and application of computer science methods/techniques, in intra/inter-disciplinary areas of **Computer Applications,** enabling to develop skills to solve problems having applications in other disciplines and/or in the real world and to formulate, synthesize, and articulate issues for analyzing, designing, and implementing of project/research proposals, testing hypotheses, and drawing inferences based on the analysis.
PLO-7: Constitutional, Humanistic, Moral Values and Ethics	Know constitutional, humanistic, moral and ethical values, and intellectual property rights to become a scholar/professional with ingrained values in expanding knowledge for the society, and to avoid unethical practices such as fabrication, falsification or misrepresentation of data or committing plagiarism.
PLO-8: Capabilities/ qualities and mindset	To exercise personal responsibility for the outputs of own work as well as of group/team and for managing complex and challenging work(s) that requires new/strategic approaches.
PLO-9: Employability and job-ready skills	Attain the knowledge and skills required for increasing employment potential, adapting to the future work and responding to the rapidly changing demands of the employers/industry/society with time, and to have strong foundation in basic and applied aspects of Computer Science so as to venture into research in different areas of computer science, jobs in scientific and various industrial sectors and/or teaching career in Computer Applications.

Chaudhary Ranbir Singh University, Jind

Scheme of Examination for Postgraduate Programme Master of Computer Applications (MCA)

as per NEP-2020 Curriculum and Credit Framework for Postgraduate Programmes
(CBCS LOCF) with effect from the session 2024-25 (in phased manner)
Framework-2
Scheme-P

Semester	Course Type	Course Code	Nomenclature of course	Theory (Th)/ Practical (P)/ Seminar/ CHM/OEC/ EEC/ Dissertation/ Project Work	Credits		Contact hours per week L: Lecture P: Practical T: Tutorial				Internal Assessment Marks	End Term Examination Marks	Total Marks	Examination hours
					Course	Sem. Total	L	T	P	Total				
1	CC-1	M24-CAP-101	Client Side Web Technology	Th	4	26	4	0	0	4	30	70	100	3
	CC-2	M24-CAP-102	Operating System & Linux	Th	4		4	0	0	4	30	70	100	3
	CC-3	M24-CAP-103	Data Structure	Th	4		4	0	0	4	30	70	100	3
	CC-4	M24-CAP-104	Object Oriented Modeling with UML.	Th	4		4	0	0	4	30	70	100	3
	PC-1	M24-CAP-105	Practical -1 (Based on CC-1 & CC-2)	P	4		0	0	8	8	30	70	100	3
	PC-2	M24-CAP-106	Practical -2 (Based on CC-3 & CC-4)	P	4		0	0	8	8	30	70	100	4
	Seminar	M24-CAP-107	Seminar	S	2		0	0	0	2	0	50	50	1

Semester	Course Type	Course Code	Nomenclature of course	Theory (Th)/ Practical (P)/ Seminar/ CHM/OEC/ EEC/ Dissertation/ Project Work	Credits		Contact hours per week L: Lecture P: Practical T: Tutorial				Internal Assessment Marks	End Term Examination Marks	Total Marks	Examination hours
					Course	Sem. Total	L	T	P	Total				
2	CC-5	M24-CAP-201	Server Side Web Technology	Th	4	26	4	0	0	4	30	70	100	3
	CC-6	M24-CAP-202	Programming in Java	Th	4		4	0	0	4	30	70	100	3
	CC-7	M24-CAP-203	Database Management Systems	Th	4		4	0	0	4	30	70	100	3
	CC-8	M24-CAP-204	Artificial Intelligence	Th	4		4	0	0	4	30	70	100	3
	PC-3	M24-CAP-205	Practical-3 (Based on CC-5 and CC-6)	P	4		0	0	8	8	30	70	100	3
	PC-4	M24-CAP-206	Practical-4 (Based on CC-7)	P	4		0	0	8	8	30	70	100	4
	CHM	M24-CHM-201	Constitutional, Human and Moral Values, and IPR	Th	2		2	0	0	2	15	35	50	3
	Internship	M24-INT-200	An internship course of 4 Credits of 4-6 weeks duration during summer vacation after 2nd semester is to be completed by every student. Internships can be either for enhancing the employability or for developing the research aptitude.								50	50	100	
3	CC-9	M24-CAP-301	Blockchain Technology	Th	4	26	4	0	0	4	30	70	100	3

Semester	Course Type	Course Code	Nomenclature of course	Theory (Th)/ Practical (P)/ Seminar/ CHM/OEC/ EEC/ Dissertation/ Project Work	Credits		Contact hours per week L: Lecture P: Practical T: Tutorial				Internal Assessment Marks	End Term Examination Marks	Total Marks	Examination hours
					Course	Sem. Total	L	T	P	Total				
	CC-10	M24-CAP-302	Machine Learning in Python	Th	4		4	0	0	4	30	70	100	3
	DEC-1 (One course is to be opted out of M24-CAP-303 to M24-CAP-305)	M24-CAP-303	Theory of Computation	Th	4		4	0	0	4	30	70	100	3
		M24-CAP-304	Computer Organisation and Architecture	Th	4		4	0	0	4	30	70	100	3
		M24-CAP-305	May be offered through MOOC/ Swayam Portal	Th	4		4	0	0	4	30	70	100	3
	DEC-2 (One course is to be opted out of M24-CAP-306 to M24-CAP-308)	M24-CAP-306	Computer Graphics	Th	4		4	0	0	4	30	70	100	3
		M24-CAP-307	Big Data & Pattern Recognition	Th	4		4	0	0	4	30	70	100	3
		M24-CAP-308	May be offered through MOOC/ Swayam Portal*	Th	4		4	0	0	4	30	70	100	3
	PC-5	M24-CAP-309	Practical-5 (Based on CC-9)	P	4		0	0	8	8	30	70	100	4

Vishal

Semester	Course Type	Course Code	Nomenclature of course	Theory (Th)/ Practical (P)/ Seminar/ CHM/OEC/ EEC/ Dissertation/ Project Work	Credits		Contact hours per week L: Lecture P: Practical T: Tutorial				Internal Assessment Marks	End Term Examination Marks	Total Marks	Examination hours
					Course	Sem. Total	L	T	P	Total				
4	PC-6	M24-CAP-310	Practical-6 (Based on CC-10)	P	4	26	0	0	8	8	30	70	100	4
	OEC	M24-OEC-308	Data Analytics using Excel	Th	2		2	0	0	2	15	35	50	3
	DEC-3 (One course is to be opted out of M24-CAP-401 to M24-CAP-404)	M24-CAP-401	Data Communication and Computer Networks	Th	4		4	0	0	4	30	70	100	3
		M24-CAP-402	Data Science	Th	4		4	0	0	4	30	70	100	3
		M24-CAP-403	Design and Analysis of Algorithm	Th	4		4	0	0	4	30	70	100	3
		M24-CAP-404	Offered through MOOC/ Swayam Portal	Th	4		4	0	0	4	30	70	100	3
	DEC-4 (One course is to be opted out of M24-CAP-405 to M24-CAP-408)	M24-CAP-405	Cyber Security	Th	4		4	0	0	4	30	70	100	3
		M24-CAP-406	Compiler Design	Th	4		4	0	0	4	30	70	100	3
		M24-CAP-407	Soft Computing	Th	4		4	0	0	4	30	70	100	3

Vishal

Semester	Course Type	Course Code	Nomenclature of course	Theory (Th)/ Practical (P)/ Seminar/ CHM/OEC/ EEC/ Dissertation/ Project Work	Credits		Contact hours per week L: Lecture P: Practical T: Tutorial				Internal Assessment Marks	End Term Examination Marks	Total Marks	Examination hours
					Course	Sem. Total	L	T	P	Total				
		M24-CAP-408	May be offered through MOOC/ Swayam Portal	Th	4		4	0	0	4	30	70	100	3
	DEC-5 (One course is to be opted out of M24-CAP-409 to M24-CAP-412)	M24-CAP-409	Mobile Computing	Th	4		4	0	0	4	30	70	100	3
		M24- CAP -410	Cloud Computing and IoT	Th	4		4	0	0	4	30	70	100	3
		M24- CAP -411	Principles of Programming Languages	Th	4		4	0	0	4	30	70	100	3
		M24- CAP -412	Offered through MOOC/ Swayam Portal*	Th	4		4	0	0	4	30	70	100	3
	EEC	M24- CAP -413	Research and Publication Ethics	Th	2		2	0	0	2	15	35	50	3
	Dissertation / Project	M24-CAP-414	Dissertation/Project	D	12		0	0	0	12	0	300	300	-

NOTES: A student can opt one elective course in a semester, i.e. up to 40% of total elective courses mentioned in the scheme, through SWAYAM/NPTEL or other online portals recognized by the UGC and the university.

- The list of MOOC will be approved by Staff Council, Department of Computer Science and Applications prior to start of the semester.

Table-1

Course composition- Theory/ Theory +Tutorial			
Course Credit	Internal Assessment marks	End term exam marks	Total marks
2	15	35	50
4	30	70	100

Table-2: Course composition- Theory + Practical

Course Credit	Theory		Practical		Total marks
Theory +Practical	Internal Assessment marks	End term exam marks	Internal Assessment marks	End term exam marks	
2+0	15	35	-		50
4+0	30	70	-	-	100
0+4	-	-	30	70	100

Table- 3: Distribution of Internal Assessment Marks (Theory)

Total Internal Assessment Marks (Theory)	Class Participation	Seminar/Presentation/Assignment/Quiz/class test, etc.	Mid-Term Exam
15	4	4	7
30	5	10	15

Table -4 Distribution of Internal Assessment Marks (Practical)

Total Internal Assessment Marks (Practicum)	Class Participation	Seminar/Demonstration/Viva-Voce/Lab record, etc.	Mid-Term Exam
30	5	10	15

Chaudhary Ranbir Singh University

(Established by the Haryana State Legislature Act 28 of 2014)
(Recognised u/s 2(f) and 12(B) of UGC Act, 1956)

# Syllabus
for

Post Graduate Programme

Master of Computer Applications

as per NEP-2020
Curriculum and Credit Framework for Postgraduate Programme

With Multiple Entry-Exit, Internship and CBCS-LOCF
With effect from the session 2024-25 (in phased manner)

DEPARTMENT OF COMPUTER SCIENCE & APPLICATIONS
FACULTY OF PHYSICAL SCIENCES

CHAUDHARY RANBIR SINGH UNIVERSITY
JIND – HARYANA – INDIA - 126102

Vishal

CC-1 Client-side Web Technology

With effect from the Session: 2024-25

Part A - Introduction			
Name of the Programme	MCA		
Semester	1^{st}		
Name of the Course	Client-side Web Technology		
Course Code	M24-CAP-101		
Course Type	CC-1		
Level of the course (As per Annexure-I	400-499		
Pre-requisite for the course (if any)			
Course Objectives	This course aims to provide a comprehensive understanding of front-end development using the MERN stack, covering HTML, CSS, and JavaScript basics. Students will learn about React for building dynamic user interfaces, including components, state management, and event handling. The course also explores advanced topics such as React Router, Redux for state management, and advanced hooks for managing side effects and context.		
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	CLO-1. Gain an understanding of the web development process and the components of the MERN stack, with a focus on HTML structure, CSS styling, and responsive design. CLO-2 Develop foundational JavaScript skills, including control structures, functions, objects, arrays, and DOM manipulation for dynamic web interactions. CLO-3 Learn the basics of React, including JSX, components, state management, lifecycle methods, and handling events and forms within React applications. CLO-4 Master advanced React topics like React Router for navigation, state management with Redux, and using advanced hooks for managing complex state and side effects.		
Credits	Theory	Practical	Total
	4	0	4
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
End Term Exam Marks	70	0	70
Max. Marks	100	0	100
Examination Time	3 hours		

Part B- Contents of the Course

Instructions for Paper- Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	**Topics**	**Contact Hours**
I	Basics of Front End Development: Overview of web development (Front End vs. Back End), Understanding the MERN stack and its components, Tools and environments (text editors, browsers, version control with Git); HTML (HyperText Markup Language): Structure of an HTML document, HTML elements and attributes, Forms and input types, Semantic HTML (header, footer, article, section, nav); CSS (Cascading Style Sheets): Basics of CSS (syntax, selectors, properties), CSS Box Model, Positioning and layout (float, flexbox, grid), Responsive design (media queries, mobile-first design).	15

I	Basics of JavaScript: Introduction to JavaScript, Variables, data types, and operators, Control structures (if, else, switch, loops); Functions and Scope: Defining and invoking functions, Function expressions and arrow functions, Scope and closures; Objects and Arrays: Creating and manipulating objects, Array methods and iteration; Regular Expressions: Introduction to RegExp, Regular expression usage, Modifiers, RegExp patterns, RegExp methods, String methods for RegExp; DOM Manipulation and Events: Selecting and manipulating DOM elements, Event handling and delegation, Creating and appending elements dynamically	15
III	Introduction to React: Overview and advantages of React, Setting up a React development environment (using Create React App); JSX (JavaScript XML): Understanding JSX syntax, Embedding expressions in JS, JSX best practices; Components and Props: Functional and class components, Props and component communication, Prop types and default props.; State and Lifecycle: Understanding state in React, State management in class components, Lifecycle methods (componentDidMount, componentDidUpdate, componentWillUnmount); Event Handling and Forms: Handling events in React, Controlled vs. uncontrolled components, Form handling and validation	15
IV	React Router: Introduction to React Router, Setting up and configuring routes, Navigating between routes and passing parameters; State Management with Redux: Introduction to Redux, Setting up Redux with React, Actions, reducers, and store, Connecting Redux to React components; Advanced Hooks: Using built-in hooks (useEffect, useContext, useReducer), Creating custom hooks, Managing side effects with useEffect	15
Total Contact Hours		60

Suggested Evaluation Methods

Internal Assessment: 30		**End Term Examination: 70**	
□ **Theory**	**30**	□ **Theory**	**70**
●Class Participation:	5	Written Examination	
●Seminar/presentation/assignment/quiz/class test etc.:	10		
●Mid-Term Exam:	15		

Part C-Learning Resources

Reference Books:

1) Flanagan, D. (2020). *JavaScript: The Definitive Guide.* O'Reilly Media.
2) Kogent Learning. (2009). *Web Technologies: HTML, JavaScript, PHP, Java, JSP, XML, AJAX – Black Book.* Wiley India Pvt. Ltd.
3) Duckett, J. (2014). *JavaScript and jQuery: Interactive Front-End Web Development.* Wiley.
4) Robson, E., & Freeman, E. (2014). *Head First JavaScript Programming: A Brain-Friendly Guide.* O'Reilly Media.
5) Banks, A., & Chinnathambi, K. (2017). *Learning React: Functional Web Development with React and Redux.* O'Reilly Media.

CC-2 Operating System and Linux

With effect from the Session: 2024-25

Part A - Introduction

Name of the Programme	MCA
Semester	1st
Name of the Course	Operating System and Linux
Course Code	M24-CAP-102
Course Type	CC-2
Level of the course (As per Annexure-I	400-499
Pre-requisite for the course (if any)	-
Course Objectives	This course provides a foundational understanding of operating systems, covering their definition, types, and functions. Students will explore system structures, process management, CPU scheduling, memory management, paging and segmentation, virtual memory, and file systems. Additionally, the course offers an introduction to Linux, including its history, architecture, file system, basic commands, shell scripting, process and user management, networking, system administration, and basic security concepts.
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	CLO-1. Understand the fundamental concepts, functions, and structures of operating systems, and apply various CPU scheduling algorithms. CLO-2 Grasp memory hierarchy, allocation techniques, paging, segmentation, virtual memory concepts, and file system management. CLO-3 Learn the history, features, and architecture of Linux, perform basic file operations, and write simple shell scripts. CLO-4 Manage processes, users, and groups in Linux, utilize network commands, perform system administration tasks, and understand basic security measures.

Credits	Theory	Practical	Total
	4	0	4
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
End Term Exam Marks	70	0	70
Max. Marks	100	0	100
Examination Time	3 hours		

Part B- Contents of the Course

Instructions for Paper- Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	**Topics**	**Contact Hours**
I	Introduction to Operating Systems: Definition, types, and functions of an operating system; System Structures: Operating system services, system calls, system programs, and system structure; Process Management: Process concept, process scheduling, operations on processes, inter-process communication; CPU Scheduling: Scheduling criteria, scheduling algorithms (FCFS, SJF, Priority, Round Robin, Multilevel Queue Scheduling).	15

	Memory Management: Memory Hierarchy, Types of memory, memory allocation techniques; Paging and Segmentation: Basic concepts, paging, segmentation, segmentation with paging; Virtual Memory: Demand paging, page replacement algorithms, allocation of frames, thrashing; File Systems: File concepts, access methods, directory and disk structure, file system mounting, file sharing, protection.	15
III	Introduction to Linux: History, features, architecture of Linux; Linux File System: File and directory structure, file permissions, standard file types; Basic Commands: File and directory operations (ls, cp, mv, rm, mkdir), text processing (cat, grep, sort), system status (ps, top, df, du); Shell Scripting: Introduction to shell, shell variables, control structures (if, case, while, for), writing simple shell scripts.	15
IV	Process Management in Linux: Managing processes (ps, top, kill, nice), job scheduling (cron, at); User and Group Management: Creating and managing users and groups, file permissions, changing ownership (chown, chgrp); Networking in Linux: Basic network commands (ifconfig, ping, netstat, ssh), configuring network interfaces; System Administration: Package management (installing and removing software using rpm, dpkg, apt-get), backup and restore, logging; Security: Basic security concepts, user authentication.	15
Total Contact Hours		60

Suggested Evaluation Methods

Internal Assessment: 30		**End Term Examination: 70**	
□ **Theory**	**30**	□ **Theory**	**70**
1) Class Participation:	5	Written Examination	
2) Seminar/presentation/assignment/quiz/class test etc.:	10		
3) Mid-Term Exam:	15		

Part C-Learning Resources

Reference Books:

1) Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). Operating System Concepts (10th ed.). Wiley.
2) Tanenbaum, A. S., & Bos, H. (2014). Modern Operating Systems (4th ed.). Pearson.
3) Stallings, W. (2018). Operating Systems: Internals and Design Principles (9th ed.). Pearson.
4) Love, R. (2013). Linux System Programming (2nd ed.). O'Reilly Media.
5) Nemeth, E., Snyder, G., Hein, T. R., & Whaley, B. (2017). UNIX and Linux System Administration Handbook (5th ed.). Pearson.
6) Sobell, M. G. (2017). A Practical Guide to Linux Commands, Editors, and Shell Programming (4th ed.). Pearson.
7) Das, S. (2012). Your UNIX/Linux: The Ultimate Guide (3rd ed.). McGraw-Hill Education.
8) Kerrisk, M. (2010). The Linux Programming Interface: A Linux and UNIX System Programming Handbook. No Starch Press.

CC-3 Data Structures

With effect from Session: 2024-25

Part A - Introduction

Name of the Programme	MCA		
Semester	1st		
Name of the Course	Data Structures		
Course Code	M24-CAP-103		
Course Type	CC-3		
Level of the course (As per Annexure-I	400-499		
Pre-requisite for the course (if any)	-		
Course Objectives	This course introduces fundamental concepts of algorithms and data structures, including algorithmic notation, programming principles, and program analysis. Students will explore arrays, searching and sorting techniques, stacks, queues, and linked lists, along with their applications. The course also covers tree structures such as binary trees, AVL trees, B-trees, and tries, as well as graph terminology, representation, and traversal methods. Additionally, students will learn about set operations, file queries, sequential organization, index techniques, and external sorting.		
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	CLO-1. Master algorithmic notation, programming principles, and implement arrays, searching and sorting techniques. CLO-2 Apply stack and queue operations, understand linked lists, and their applications including dynamic storage management. CLO-3 Comprehend binary trees, binary search trees, AVL trees, B-trees, B+ tree indexing, Trie tree indexing, and their applications. CLO-4 Utilize graph representations, traversals, applications, sets operations, and file organization techniques.		
Credits	Theory	Practical	Total
	4	0	4
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
End Term Exam Marks	70	0	70
Max. Marks	100	0	100
Examination Time	3 hours		

Part B- Contents of the Course

Instructions for Paper- Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	**Topics**	**Contact Hours**
I	Introduction: Algorithmic notation – Programming principles – Creating programs-Analyzing programs. Arrays: One dimensional array, multidimensional array, pointer arrays. Searching: Linear search, Binary Search, Fibonacci search. Sorting techniques: Internal sorting - Insertion Sort, Selection Sort, Shell Sort, Bubble Sort, Quick Sort, Heap Sort, Merge Sort and Radix Sort.	15
II	Stacks: Definition – operations - applications of stack. Queues: Definition - operations - Priority queues – Dequeues – Applications of queue. Linked List: Singly Linked List, Doubly Linked List, Circular Linked List, linked stacks, Linked queues, Applications of Linked List – Dynamic storage management – Generalized list.	15
III	Trees: Binary tree, Terminology, Representation, Traversals, Applications – Binary search tree – AVL tree. B Trees: B Tree indexing, operations on a B Tree, Lower and upper bounds of a B Tree - B + Tree Indexing – Trie Tree Indexing.	15

Graph: Terminology, Representation, Traversals – Applications - spanning trees, shortest path and Transitive closure, Topological sort. Sets: Representation - Operations on sets – Applications. Files: queries - Sequential organization – Index techniques. External sorting.	15
Total Contact Hours	60

Suggested Evaluation Methods			
Internal Assessment: 30		**End Term Examination: 70**	
□ **Theory**	**30**	□ **Theory**	**70**
●Class Participation:	5	Written Examination	
●Seminar/presentation/assignment/quiz/class test etc.:	10		
●Mid-Term Exam:	15		

Part C-Learning Resources

Reference Books:

1) Horowitz, E., & Sahni, S. (2004). *Fundamentals of Data Structures*. Galgotia Book Source Pvt. Ltd.
2) Samanta, D. (2012). *Classic Data Structures* (2nd ed.). Prentice-Hall of India Pvt. Ltd., India.
3) Kruse, R., Tondo, C. L., & Leung, B. (2007). *Data Structures and Program Design in C* (2nd ed.). Prentice-Hall of India Pvt. Ltd.
4) Weiss, M. A. (2006). *Data Structures and Algorithm Analysis in C* (2nd ed.). Pearson Education.

CC-4 Object Oriented Modeling with UML

With effect from Session: 2024-25

Part A - Introduction

Name of the Programme	MCA
Semester	1st
Name of the Course	Object Oriented Modeling with UML
Course Code	M24-CAP-104
Course Type	CC-4
Level of the course (As per Annexure-I	400-499
Pre-requisite for the course (if any)	-
Course Objectives	This course provides a comprehensive introduction to Object Oriented Design with Unified Modelling Language, covering its history, features, and applications. Students will learn Object Oriented basics, including object model, class diagram, State Transition Diagram, Abstraction, Encapsulation, Inheritance, Polymorphism, Scripts etc..
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	CLO-1. Understand Object Oriented Modeling and UML background, features. CLO-2 Master object-oriented programming principles including classes, objects, inheritance, polymorphism, encapsulation, abstraction etc. CLO-3 Gain Proficiency in Object Oriented Modeling and UML for Software Design purpose. CLO-4 Explore and utilize advanced features of Object Oriented Modeling and UML for Use Case Diagram.

Credits	Theory	Practical	Total
	4	0	4
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
End Term Exam Marks	70	0	70
Max. Marks	100	0	100
Examination Time	3 hours		

Part B- Contents of the Course

Instructions for Paper- Setter: The examiner will set 9 questions asking two questions from each unit and one compulsory question by taking course learning outcomes (CLOs) into consideration. The compulsory question (Question No. 1) will consist at least 4 parts covering entire syllabus. The examinee will be required to attempt 5 questions, selecting one question from each unit and the compulsory question. All questions will carry equal marks.

Unit	**Topics**	**Contact Hours**
I	Introduction: Object-Orientation, Object Oriented Methodology, Modeling, Class Modeling: Object, Class, Value & Attributes, Operation & Method, Link & Association, Association Classes, Qualified association, Multiplicity, Association end name, Ordering, Bag &Sequences, Generalization & Inheritance, Uses of Generalization.	15
II	Advance Class Modeling: Advanced Object & Class Concepts, N-Array association, Aggregation, Abstract Class, Multiple Inheritance, Metadata. State Modeling: Events, States, Transition & Conditions, State Diagram, State Diagram Behavior. Advanced State Modeling: Nested State Diagram, Nested States, Signal Generalization, Concurrency.	15

	System Design: Overview, Estimating Performance, Making a reuse plan, Breaking a system into subsystems, Identifying Concurrency, Allocation of subsystem, Management of data storage, Handling global resources, Choosing a Software control strategy, Handling boundary conditions, Setting trade-off priorities.	15
IV	Use Case Models: Actors, Use case, Use case diagram, Guidelines for use case diagram. Sequence Model: Scenarios, Sequence Diagrams, Guidelines for Sequence model. Activity Model: Activities, Branches, Initiation & Termination, Concurrent Activities, Executable Activity Diagram, Guidelines for Activity diagram. Case Study: Working of ATM with reference to implementation of basic structure, advanced structure, and functionality.	15
Total Contact Hours		60

Suggested Evaluation Methods

Internal Assessment: 30		**End Term Examination: 70**	
□ **Theory**	30	□ **Theory**	70
● Class Participation:	5	Written Examination	
● Seminar/presentation/assignment/quiz/class test etc.:	10		
● Mid-Term Exam:	15		

Part C-Learning Resources

Reference Books:

1) Michael Blaha, James Rumbaugh, *"Object Oriented Modeling and Design with UML"*, Pearson Education, 2011.
2) Daminni Grover, *"Object Oriented Analysis and Design with UML"*, I. K International Publishing House, 1st edition, 2012.
3) Martin Fowler, *"UML Distilled"*, Pearson Education Inc., 2018.
4) Mike O'Docherty, *"Object Oriented Analysis And Design Understanding System Development with UML 2.0"*, Wiley Dreamtech, 2005.

PC-1 PRACTICAL-1 (Based on CC-1 & CC-2)

With effect from Session: 2024-25

Part A - Introduction

Name of the Programme	MCA
Semester	Ist
Name of the Course	Practical-1
Course Code	M24-CAP-105
Course Type	PC-1
Level of the course	400-499
Pre-requisite for the course (if any)	
Course objectives	This is a laboratory course and the objective of this course is to acquaint the students with the understanding and implementing of client-side web technologies. Also, the concepts of operating systems and shell programming will be implemented by the students.
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	CLO 1: Solve practical problems related to theory courses undertaken in the CC-1 and CC-2 from application point of view. CLO 2: Know how to use the client-side web technologies. CLO 3: implement the various functions of operating systems. CLO 4: Designing and implementing the shell programs in Linux.

Credits	Theory	Practical	Total
	0	4	4
Teaching Hours per week	0	8	8
Internal Assessment Marks	0	30	30
End Term Exam Marks	0	70	70
Max. Marks	0	100	100
Examination Time	0	4 hours	

Part B- Contents of the Course

Practicals	Contact Hours
Practical course will consist of two components Part-A and Part-B. The examiner will set 5 questions at the time of practical examination asking 2 questions from the Part-A and 3 questions from the Part-B by taking course learning outcomes (CLO) into consideration. The examinee will be required to solve one problem from the Part-A and to write and execute 2 questions from the Part-B.	120

Part-A

HTML/CSS Basics:

- Creating a webpage structure with HTML.
- Styling the webpage using CSS (inline, internal, and external styles).

□ **Responsive Design:**

- Making the webpage responsive using media queries.
- Using frameworks like Bootstrap for responsive design.

□ **JavaScript Basics:**

- Adding interactivity with JavaScript (DOM manipulation, event handling).
- Working with variables, loops, and conditions.

Frameworks and Libraries:

- Using front-end frameworks React.
- Utilizing libraries such as jQuery for DOM manipulation.

□ **Introduction to React:**

- Create a simple React component that displays "Hello, World!" on the screen.
- Use JSX syntax and explain its advantages over plain JavaScript.

□ **State and Props:**

- Build a component that takes props and renders them.
- Implement state in a component and update it based on user interaction (e.g., button click).

□ **Basic Todo App:**

Develop a Todo application where users can add, delete, and mark tasks as completed.

Use state to manage the list of tasks.

Using React Router:

- Set up React Router in a project and create multiple pages (e.g., Home, About, Contact).
- Implement navigation between these pages using Link and NavLink.

□ **Redux Integration:**

- Integrate Redux for state management in a React application.
- Implement actions, reducers, and connect components to Redux store.

□ **Responsive Design with React Router:**

- Build a responsive multi-page application using React Router.
- Ensure layout adjustments for different screen sizes using CSS media queries or frameworks like Bootstrap.

Part-B	60 (Lab hours include instructions for writing programs and demonstration by a teacher and for running the programs on computer by students.)

1) Implement a simple program demonstrating the creation and synchronization of threads or processes.
2) Design and simulate a memory management system (e.g., paging, segmentation).
3) Implement algorithms like First Fit, Best Fit, and Worst Fit for memory allocation.
4) Implement a basic file system with operations like file creation, deletion, reading, and writing.
5) Compare different file allocation methods (e.g., contiguous allocation, linked allocation, indexed allocation).
6) Solve synchronization problems such as the producer-consumer problem or dining philosophers problem using semaphores or mutexes.
7) Implement a solution for deadlock prevention, avoidance, or detection.
8) Profile and analyze the performance of different scheduling algorithms (e.g., FCFS, SJF, Round Robin) using simulations.
9) Evaluate the impact of caching and paging strategies on system performance.
10) Write a shell script named hello.sh that prints "Hello, World!" to the terminal when executed.
11) Demonstrate running the script and explain how to make it executable using chmod.
12) Write a script greet_user.sh that prompts the user for their name and then prints a personalized greeting.
13) Use variables to store user input and demonstrate the use of read command.
14) Create a script check_number.sh that accepts a number as an argument.
15) Check if the number is positive, negative, or zero, and print an appropriate message using conditional statements (if-else).
16) Develop a script countdown.sh that takes a number as input and prints a countdown from that number to 1.
17) Use a loop (e.g., while or for) to implement the countdown.
18) Write a script file_info.sh that accepts a filename as an argument.
19) Check if the file exists and whether it is a regular file or directory. Display appropriate messages based on the checks.
20) Create a script word_count.sh that reads a text file (provided as an argument) and counts the number of words in the file.
21) Utilize command-line tools like wc and cat for reading and counting words.

Suggested Evaluation Methods

Internal Assessment: 30		**End Term Examination: 70**
□ **Practicum**	30	□ **Practicum** 70
•Class Participation:	5	Lab record, Viva-Voce, write-up and execution of the programs
•Seminar/Demonstration/Viva-voce/Lab records etc.:	10	
•Mid-Term Examination:	15	

Part C-Learning Resources

commended Books/e-resources/LMS:

1) Flanagan, D. (2020). *JavaScript: The Definitive Guide*. O'Reilly Media.
2) Kogent Learning. (2009). *Web Technologies: HTML, JavaScript, PHP, Java, JSP, XML, AJAX – Black Book*. Wiley India Pvt. Ltd.
3) Duckett, J. (2014). *JavaScript and jQuery: Interactive Front-End Web Development*. Wiley.
4) Robson, E., & Freeman, E. (2014). *Head First JavaScript Programming: A Brain-Friendly Guide*. O'Reilly Media.
5) Banks, A., & Chinnathambi, K. (2017). *Learning React: Functional Web Development with React and Redux*. O'Reilly Media.
6) Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). Operating System Concepts (10th ed.). Wiley.
7) Tanenbaum, A. S., & Bos, H. (2014). Modern Operating Systems (4th ed.). Pearson.
8) Stallings, W. (2018). Operating Systems: Internals and Design Principles (9th ed.). Pearson.
9) Love, R. (2013). Linux System Programming (2nd ed.). O'Reilly Media.
10) Nemeth, E., Snyder, G., Hein, T. R., & Whaley, B. (2017). UNIX and Linux System Administration Handbook (5th ed.). Pearson.
11) Sobell, M. G. (2017). A Practical Guide to Linux Commands, Editors, and Shell Programming (4th ed.). Pearson.
12) Das, S. (2012). Your UNIX/Linux: The Ultimate Guide (3rd ed.). McGraw-Hill Education.
13) Kerrisk, M. (2010). The Linux Programming Interface: A Linux and UNIX System Programming Handbook. No Starch Press

PC-2 PRACTICAL-2 (Based on CC-3 & CC-4)

With effect from Session: 2024-25

Part A - Introduction

Name of the Programme	MCA
Semester	Ist
Name of the Course	Practical-2
Course Code	M24-CAP-106
Course Type	PC-2
Level of the course	400-499
Pre-requisite for the course (if any)	
Course objectives	This is a laboratory course and the objective of this course is to acquaint the students with the understanding and implementation of various data structures. Also, the students will implement the concepts of programming with Java.
Course Learning Outcomes (CLO). After completing this course, the learner will be able to:	CLO 1: Solve practical problems related to theory courses undertaken in the CC-3 and CC-4 from an application point of view. CLO 2: Know how to use and implement the various data structures. CLO 3: Implement the various features of Java Programming by writing suitable programs. CLO 4: Designing and implementing applications in Java.

	Theory	Practical	Total
Credits	0	4	4
Teaching Hours per week	0	8	8
Internal Assessment Marks	0	30	30
End Term Exam Marks	0	70	70
Max. Marks	0	100	100
Examination Time	0	4 hours	

Part B- Contents of the Course

Practicals	Contact Hours
Practical course will consist of two components Part-A and Part-B. The examiner will set 5 questions at the time of practical examination asking 2 questions from the Part-A and 3 questions from the Part-B by taking course learning outcomes (CLO) into consideration. The examinee will be required to solve one problem from the Part-A and to write and execute 2 questions from the Part-B.	120

Part-A

60

Task 1: Linked List Implementation

- Implement a singly linked list in a programming language of your choice (e.g., C/C++, Java, Python).
- Include functions/methods for insertion (at the beginning, end, and specific position), deletion, and traversal.

□ Task 2: Stack Operations

- Implement a stack using an array or linked list.
- Include functions/methods for push, pop, peek, and checking if the stack is empty or full.

□ Task 3: Queue Implementation

- Implement a queue using an array or linked list.
- Include functions/methods for enqueue, dequeue, peek, and checking if the queue is empty or full.

□ Task 4: Binary Search Tree (BST) Operations

- Implement a binary search tree (BST) in your chosen programming language.
- Include functions/methods for insertion, deletion, searching for a key, finding minimum and maximum values, and traversing the tree (inorder, preorder, postorder).

□ Task 6: Sorting Algorithms

- Implement at least two sorting and searching algorithms (e.g., selection sort, insertion sort, merge sort, quick sort).
- Compare their time complexity and performance using different input sizes.

□ Task 7: Graph Representation and Algorithms

- Implement an adjacency list representation of a graph.
- Include functions/methods for BFS (Breadth-First Search) and DFS (Depth-First Search) traversal of the graph.

Part-B

1) Write a Java program that converts temperatures between Celsius and Fahrenheit based on user input using methods for conversion and input validation.
2) Implement a Java program to perform matrix addition, multiplication, and transpose operations using arrays and methods.
3) Develop a Java program that converts a decimal number to its binary, octal, and hexadecimal equivalents using loops and methods.
4) Create a Java program to simulate a simple bank account management system with features like deposit, withdrawal, and balance inquiry using classes, objects, and encapsulation.
5) Write a Java program that reads a text file, counts the occurrences of each word, and displays the top N most frequent words using HashMap for storage and sorting.
6) Implement a Java program to generate the first N prime numbers using a combination of loops, methods, and optimizations like the Sieve of Eratosthenes algorithm.
7) Develop a Java program that takes a month and year as input and prints the calendar for that month using control flow statements and loops for date calculation.
8) Write a Java program that generates different number patterns like pyramid patterns using nested loops and methods for pattern printing.
9) Create a Java program to manage an employee payroll system with features for adding employees, calculating salaries based on hours worked or monthly salary, and generating pay slips using classes, inheritance, and polymorphism.
10) Implement Java programs to compare the performance of different sorting algorithms (like quicksort, mergesort, and heapsort) on large arrays of integers, measuring and analyzing time complexity.
11) Develop a Java program that recursively searches a directory for files matching a given pattern and displays the file paths using recursion and file handling classes.
12) Write a Java program to perform arithmetic operations (addition, subtraction, multiplication, division) on large numbers using BigInteger class and exception handling for division by zero.
13) Implement a Java program to solve the Tower of Hanoi problem for N disks using recursion, demonstrating the steps and movements required.
14) Write a Java program to find the largest and smallest elements in an array.
15) Implement a Java program to sort an array of integers using bubble sort.
16) Create a Java program to find the frequency of each element in an array.
17) Develop a Java program to reverse an array without using an additional array.
18) Write a Java program to merge two sorted arrays into a single sorted array.
19) Define a Java class representing a Student with private instance variables and public getter and setter methods.
20) Create a Java program to demonstrate constructor overloading in a class.
21) Implement a Java program to calculate the area and perimeter of a rectangle using a class and object.
22) Develop a Java program to implement inheritance by creating a base class Animal and derived classes like Dog and Cat.
23) Write a Java program to demonstrate method overriding by implementing a base class Shape and derived classes like Circle and Rectangle.

60
(Lab hours include instructions for writing programs and demonstration by a teacher and for running the programs on computer by students.)

Suggested Evaluation Methods

Internal Assessment: 30		End Term Examination: 70	
□ Practicum	30	□ Practicum	70
•Class Participation:	5	Lab record, Viva-Voce, write-up and execution of the programs	
•Seminar/Demonstration/Viva-voce/Lab records etc.:	10		
•Mid-Term Examination:	15		

Part C-Learning Resources

ecommended Books/e-resources/LMS:

1) Horowitz, E., & Sahni, S. (2004). *Fundamentals of Data Structures.* Galgotia Book Source Pvt. Ltd.
2) amanta, D. (2012). *Classic Data Structures* (2nd ed.). Prentice-Hall of India Pvt. Ltd., India.
3) Kruse, R., Tondo, C. L., & Leung, B. (2007). *Data Structures and Program Design in C* (2nd ed.). Prentice-Hall of India Pvt. Ltd.
4) Weiss, M. A. (2006). *Data Structures and Algorithm Analysis in C* (2nd ed.). Pearson Education.
5) Balaguruswamy, E. (2009). *Programming with JAVA: A Primer.* Tata McGraw Hill.
6) Naughton, P., & Schildt, H. (2002). *The Complete Reference Java 2.* Tata McGraw Hill.
7) Neimeyer, P., & Peck, J. (1996). *Exploring Java.* O'Reilly.
8) Hahn, H. (1996). *Teach Yourself the Internet.* Prentice-Hall of India (P.H.I.).
9) Boone, B., & Stanek, W. (2001). *Java 2 Exam Guide.* Tata McGraw Hill.

Department of Computer Science and Applications
Chaudhary Ranbir Singh University, Jind

Eligibility

Master of Computer Applications (2 years)

A 3-year/6-semester Bachelor's degree with at least 50% marks (47.5% marks in case of candidates belonging to reserved category: SC/ST/Differently abled) with a minimum of 120 credits for Master of Computer Applications (2 years)

(For students having no Mathematics background at 10+2 level or at graduation level, a compulsory bridge course, framed by the University is required to be passed. Besides, the students have to undergo an additional bridge course related to computer subjects for students having no Computer Science background at 10+2 level or at graduation level. Students who have no background of Computer Science in Graduation but successfully completed Post Graduate Diploma in Computer Science / Computer Applications / Data Science / Artificial Intelligence are exempted from bridge course of Computer Science).

Master of Computer Applications (1 year)

B Tech/B.E. (CSE/IT/AI/ML/DS/AI&ML/AI&DS) or equivalent or Bachelor's degree with Honours/ Honours with Research with Computer Science or Computer Applications as a major course or BCA with Honours/ Honours with Research (with at least 50% marks; 47.5% marks in case of candidates belonging to reserved category: SC/ST/Differently abled) with a minimum of 160 credits for Master of Computer Applications (1 year)

Note : This eligibility will be rolled out progressively.

10/07/2024