ANX - C

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)

Scheme and Syllabus
for
Post Graduate Diploma in Computer Applications

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

With Internship and CBCS-LOCF
With effect from Academic Session 2025-26

DEPARTMENT OF COMPUTER SCIENCE AND APPLICATIONS
FACULTY OF PHYSICAL SCIENCES

CHAUDHARY RANBIR SINGH UNIVERSITY, JIND – 126102

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	

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	PGD24-CAP-101	Client Side Web Technology	Th	4	26	4	0	0	4	30	70	100	3
	CC-2	PGD24-CAP-102	Operating System & Linux	Th	4		4	0	0	4	30	70	100	3
	CC-3	PGD 24-CAP-103	Data Structure	Th	4		4	0	0	4	30	70	100	3
	CC-4	PGD 24-CAP-104	Object Oriented Modeling with UML.	Th	4		4	0	0	4	30	70	100	3
	PC-1	PGD 24-CAP-105	Practical -1 (Based on CC-1 & CC-2)	P	4		0	0	8	8	30	70	100	3
	PC-2	PGD 24-CAP-106	Practical -2 (Based on CC-3 & CC-4)	P	4		0	0	8	8	30	70	100	4
	Seminar	PGD 24-CAP-107	Seminar	S	2		0	0	0	2	0	50	50	1
2	CC-5	PGD24-CAP-201	Server Side Web Technology	Th	4	26	4	0	0	4	30	70	100	3
	CC-6	PGD 24-CAP-202	Programming in Java	Th	4		4	0	0	4	30	70	100	3
	CC-7	PGD 24-CAP-203	Database Management Systems	Th	4		4	0	0	4	30	70	100	3
	CC-8	PGD24-CAP-204	Artificial Intelligence	Th	4		4	0	0	4	30	70	100	3
	PC-3	PGD24-CAP-205	Practical-3 (Based on CC-5 and CC-6)	P	4		0	0	8	8	30	70	100	3
	PC-4	PGD24-CAP-206	Practical-4 (Based on CC-7)	P	4		0	0	8	8	30	70	100	4
	CHM	PGD24-CHM-201	Constitutional, Human and Moral Values, and IPR	Th	2		2	0	0	2	15	35	50	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	Exam hours
					Course	Sem. Total	L	T	P	Total				
	Internship	PGD24-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	

With effect from the Session: 2025-26

Part A - Introduction

Name of the Programme	PGDCA
Semester	1st
Name of the Course	Client-side Web Technology
Course Code	PGD25-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

II	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**
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) 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.

With effect from the Session: 2025-26

Part A - Introduction

Name of the Programme	PGDCA
Semester	1st
Name of the Course	Operating System and Linux
Course Code	PGD25-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.

	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

II	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.

With effect from Session: 2025-26

Part A - Introduction

Programme	PGDCA
Semester	1st
Name of the Course	Data Structures
Course Code	PGD25-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

IV	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**
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) 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.

With effect from Session: 2025-26

Part A - Introduction

Name of the Programme	PGDCA
Semester	1st
Name of the Course	Object Oriented Modeling with UML
Course Code	PGD25-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.

	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

III	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**
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) 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: 2025-26

Part A - Introduction

Name of the Programme	PGDCA
Semester	1st
Name of the Course	Practical-1
Course Code	PGD25-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:

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

Responsive Design:

1) Making the webpage responsive using media queries.
2) Using frameworks like Bootstrap for responsive design.

JavaScript Basics:

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

Frameworks and Libraries:

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

Introduction to React:

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

State and Props:

1) Build a component that takes props and renders them.
2) 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:

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

Redux Integration:

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

Responsive Design with React Router:

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

Part-B	
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.	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**
1) Class Participation:	5	Lab record, Viva-Voce, write-up and execution of the programs	
2) Seminar/Demonstration/Viva-voce/Lab records etc.:	10		
3) Mid-Term Examination:	15		

Part C-Learning Resources

Recommended 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: 2025-26

Part A - Introduction

Name of the Programme	PGDCA
Semester	I[st]
Name of the Course	Practical-2
Course Code	PGD25-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.

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 write and execute 2 questions from the Part-B.	120

Part-A

Task 1: Linked List Implementation

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

Task 2: Stack Operations

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

Task 3: Queue Implementation

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

Task 4: Binary Search Tree (BST) Operations

1) Implement a binary search tree (BST) in your chosen programming language.
2) 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

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

Task 7: Graph Representation and Algorithms

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

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) 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.

Suggested Evaluation Methods

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

Part C-Learning Resources

Recommended Books/e-resources/LMS:

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.
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.

With effect from the Session: 2024-25

Part A - Introduction

Name of the Programme	PGDCA
Semester	2nd
Name of the Course	Server Side Web Technology
Course Code	PGD24-CAP-201
Course Type	CC-5
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 back-end development using the Node JS, Dot NET, C#, PHP and Ruby basics. Students will learn about Node JS and Dot NET for building dynamic server side scripting and programming, including components, state management, and event handling. The course also explores advanced topics such as PHP and Ruby for 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 Node JS. CLO-2 To learn PHP for development of Dynamic Websites. CLO-3 To learn .NET to provide a runtime environment and a set of libraries and tools for building and running applications on Windows operating systems. CLO-4 To learn Ruby for DevOps, automation and Website Deployment

Credits	Theory	Practical	Total
	4	0	4
Teaching Hours per week	4	0	4
Internal Assessment Marks	30	0	30
Term Exam Marks	70	0	70
Total 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	Node JS: Introduction to Node JS, REPL Terminal, Node.js Modules, Module Types, Core Modules, Local Modules, Module Experts Node Packet Manager (NMP), Installing Packages Locally, Adding dependency in Packages, Installing Packages Globally, Updating packages. Creating Web Server, Handling http requests, sending requests. Buffers, Streams, Files, reading, writing, updating files, synchronous and asynchronous. Events in Node JS, significance of the events, writing own events, Event Emitter class, inhering events. Express framework to create web applications: Configuring Routes, Working with Express. How to serve Static HTML pages to the browser, and serving other file formats and restricting certain files. Database Connectivity: MySQL Database, Creating Connection	15

II	Dot NET: The .Net framework: Introduction, The Origin of .Net Technology, Common Language Runtime (CLR), Common Type System (CTS), Common Language Specification (CLS), Microsoft Intermediate Language (MSIL), Just-In –Time Compilation, Framework Base Classes. C -Sharp Language (C#): Introduction, Data Types, Identifiers, Variables, Constants, Literals, Array and Strings, Object and Classes, Inheritance and Polymorphism, Operator Overloading, Interfaces, Delegates and Events. Type conversion. C# Using Libraries: Namespace- System, Input-Output, Multi-Threading, Networking and sockets, Managing Console I/O Operations, Windows Forms, Error Handling. .Net Assemblies and Attribute: .Net Assemblies features and structure, private and share assemblies, Built-In attribute and custom attribute. Introduction about generic.	15
III	PHP: Variables, Constant, Data types, Operator and Expression Handling, Making Decisions, Iterations, Function, Call by value, Call by reference, Recursive function, Strings, Arrays, Index based and Associative array, each() and foreach(), Working with file and Directories, State management , Regular expression, Pattern matching, Introduction to OOPS, Objects, Class, Constructor Destructor, $this variable, Public ,private, protected properties, Inheritance, code reusability, Polymorphism Parent:: & self:: keyword, Instance of operator, Abstract method and class Interface, Final Exception Handling Understanding Exception and error Try, catch, throw.	15
IV	Ruby: Introduction to Ruby, Ruby Command Line Option, Ruby Environment Variables, Whitespace, Identifier, Reserve Words, Begin, End Statement, Comments, Creating Objects, Member Functions, Variables, Constants and Literals, Operators, Conditional Control, Loop Control, Methods, Blocks, Modules and Mixins, Strings, Arrays, Hashes, Ranges, Iterators, File I/O, Exceptions, DBI, Web Applications.	15
Total Contact Hours		60

Suggested Evaluation Methods

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

Part C-Learning Resources

Reference Books:

6) Mario Casciario (2020). *Node.JS Design Patterns*. Packt Publishers.
7) David Harron (2020). *Node.JS Web Development*. Packt Publishers.
8) Vikram Vaswani (2017). *PHP – A Beginner's Guide*. McGraw Hill Education (Indian Edition)
9) Mark J. Price (2023). *C# 12 and .NET 8*. Packt Publishers.
10) Noel Rappin, Dave Thomas (2024). *Programming Ruby 3.3*. Pragmatic Bookshelf.

With effect from the Session: 2024-25

Part A - Introduction

Name of the Programme	PGDCA
Semester	2nd
Name of the Course	Programming in Java
Course Code	PGD24-CAP-202
Course Type	CC-6
Level of the course (As per Annexure-I	400-499
Pre-requisite for the course (if any)	-
Course Objectives	This course provides a sound foundation to the students on the concepts, precepts and practices, in a field that is of immense concern to the industry and business. It will cover the need to explore advanced topic of Java programming for solving problems, Design and develop GUI applications using Swings, Enhance knowledge to manipulate and store data, To provide foundations on Java Beans, Struts and JSON.
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	CLO-1 To learn why Java is useful for the design of desktop and web applications. CLO–2 To learn how to implement object-oriented designs with Java. CLO-3 To identify Java language components and how they work together in applications. CLO-4 To design and program stand-alone Java applications. CLO-5 To learn how to design a graphical user interface (GUI) with Java Swing. CLO-6 To understand how to use Java APIs for program development. CLO-7 To learn how to extend Java classes with inheritance and dynamic binding. CLO-8 To learn how to use exception handling in Java applications. CLO-9 To understand how to design GUI components with the Java Swing API. CLO-10 To learn Java generics and how to use the Java Collections API. CLO-11 To understand how to design applications with threads in Java. CLO-12 To learn how to read and write files in Java.

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: Java Features, Java Virtual Machine (JVM), Byte code, Java API, Java Development Kit (JDK), Garbage Collection. Language Basics: Keywords, Constants, Variables and Data Types, Operators and Expressions, Decision Making , Branching and Looping. Introducing Classes, Objects and Methods: Defining a Class, Methods Declaration, Creating Objects and accessing Class members, Constructors, Methods Overloading, Wrapper Classes, Inheritance, Methods Overriding, Final Class, variables and methods, Abstract Class and Methods, Interfaces.	15
II	Arrays, Strings and Vectors: Creating and using Arrays, String operations, String Buffer, String Builder, and StringTokenizer class, Vector class. Packages and Exceptions: Java API packages, Creating and using packages, static import, Exceptions handling, Types of Exceptions, multiple catch statements, 'throw' and 'throws', using 'finally' statement, Creating your own exceptions.	15
III	Multithreaded Programming: Single threaded and multi-threaded program, Creating threads using Thread class, Life cycle of a Thread, Stopping and blocking a Tread, getting and setting the Thread Priority, Synchronization, implementing the Runnable interface. Managing Input/Output Streams: Concept of streams, Byte and Character streams, Reading and Writing from Console and Files. Input output exceptions.	15
IV	Applet Programming: How Applets differs from Java Application, Applet Life Cycle, APPLET Tag, Running an Applet, Passing Parameters to Applet. Event Handling: Mechanism, The Delegation Event Model, Event Classes, Event Listener Interfaces, Adapter and inner classes. GUI Programming: Working with Frame Window, Graphics and Text, AWT Controls and classes. Layout Managers, working with Menus.	15
Total Contact Hours		60

Suggested Evaluation Methods

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

Part C-Learning Resources

Reference Books:

1. E. Balagurusamy, "*Programming with Java :A Primer*", McGraw Hill, 3rd edition.
2. Herbert Schildt, "*Java:The Complete Reference*", McGraw Hill, 7th edition.
3. Bruce Eckel, "*Thinking in Java*", Prentice Hall, 4th Edition.
4. Cay S. Horstmann, Gary Cornell, "*Core Java Volume I—Fundamentals*", Prentice Hall, 9th Edition.

With effect from Session: 2024-25

Part A - Introduction

Name of the Programme	PGDCA
Semester	2nd
Name of the Course	Data Base Management Systems
Course Code	PGD24-CAP-203
Course Type	CC-7
Level of the course (As per Annexure-I	400-499
Pre-requisite for the course (if any)	-
Course Objectives	The objective of the course is to present an introduction to database management systems, with an emphasis on how to organize, maintain and retrieve - efficiently, and effectively - information from a DBMS. This course ensures to develop a foundational understanding of MongoDB, covering its principles, architecture, and essential operations. You'll gain hands-on skills installing MongoDB, executing CRUD operations, and navigating its architecture
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	CLO-1. Describe the fundamental elements of relational database management systems. CLO-2 Explain the basic concepts of relational data model, entity-relationship model, relational database design, relational algebra and SQL. CLO-3 Learn hands on skill training of MongoDB. CLO-4 Execute CRUD Operations and navigating architecture, design schema in MongoDB.

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.

	Topics	Contact Hours
I	Database System: Definition, Characteristics, Relational data models, Schemas, Instances, Three schema architecture and data independence. Data modeling : Entity, Entity type, Entity set, Attributes, Keys, Relationship types, Relationship Sets, Roles and Structural Constraints, Weak entity type, Naming convention for ER diagram, Design issues, Subclass, Super class, Inheritance, Specialization & Generalization.	15
II	Relational Algebra: Select, Project, Join, Division, Union, Intersection, Minus, Cartesian product. Relational Calculus: Tuple variable, Range relations, Expressions, Formulas, Existential & Universal Quantifiers & there transformation, Using the Universal Quantifier, Safe expression, Domain relational calculus.	15

III	Transaction Processing: Introduction, Single User, Multiuser, Read and Write Operation, Lost update problem, Temporary update, Incorrect summary problem, Traction states, System log, Commit point of a transaction, Desirable properties, Serial, Non serial & Conflict Serializable schedule, Testing of Conflict Serializability of a schedule, View equivalence & View serializability.	15
IV	Introduction to NoSQL. Classification, Features, Advantages, Disadvantages, Types of NoSQL, RDBMS vs NoSQL Database. Getting Started with MongoDB, 2dSphere Index, Aggregation, Authentication Mechanism in MongoDB, Backing Up and Restoring Data, Bulk Operations, Collections, Configuration, CRUD Operations, Mongo as Replica Set, Mongo as Shards, Pluggable Storage Engines, Querying for Data, Replication, Update Operators, Java Drivers, Python Drivers, Upserts and Inserts.	15
Total Contact Hours		60

Suggested Evaluation Methods

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

Part C-Learning Resources

Reference Books:

1. Ramez Elmasri, Shamkant B. Navathe, *"Fundamentals of Database Systems"*, Pearson, 6th edition, 2010.
2. Silberschatz Abraham , *"Database System Concept"*, Tata Mc Graw Hill, 7th edition, 2019.
3. C. J Date , *"Introduction to Database Systems"*, Pearson Education, 8th edition, 2004.
4. Krishnan Ram and Gehrke , *"Database Management System"*, , Tata Mc Graw Hill, 2003.
5. Byross Ivan , *"Oracle 10 G The Database with HTML Database"*, BPB publication,2006.
6. Marko Eleksendric et.al., " Mastering MongoDB 7.0", Fourth Edition, Packt Publishers, 2024

With effect from Session: 2024-25

Part A - Introduction

Name of the Programme	PGDCA
Semester	2nd
Name of the Course	Artificial Intelligence
Course Code	PGD24-CAP-203
Course Type	CC-7
Level of the course (As per Annexure-I	400-499
Pre-requisite for the course (if any)	-
Course Objectives	This course provides a develop software solutions demonstrating intelligent behaviour, handling uncertainty, constantly learning, and effectively using domain knowledge. To promote research in intelligent technology and concepts. To participate in life-long learning for effective professional growth and demonstrate leadership qualities to co-ordinate cooperative team in contributing for the betterment of the society.
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	CLO-1. To learn about Heuristic Search Techniques. CLO-2 To apply propotional and predicate logic. CLO-3 To gain proficiency in Knowledge Ontologies. CLO-4 To learn about neural network and neural computing. CLO-5 To implement Fuzzy Logic and Fuzzy Arithmetic.

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
Total 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 AI: Definition of AI, History, Domains AI, AI problems & State space, Some examples problems representations like Travelling Salespersons, Syntax analysis Problem, Basic issues to solve AI problems, Underlying assumptions, AI techniques, Level of model, Criteria for success, Control strategies. Searching Techniques: DFS, BFS, Heuristic Search Techniques: Generate & Test: Hill Climbing (simple & steepest), Best first search/A*, Problem Reduction/AO*, Constraint satisfaction, Alpha-Beta pruning.	15
II	Reasoning in logic : Brief revision of propositional and predicate logic. Different characterizations of reasoning. Generalized modus ponens. Resolution. Forward and backward chaining. Knowledge Representation, Diversity of knowledge. Inheritance hierarchies. Semantic networks. Knowledgebase ontologies. Handling uncertainty, Diversity of uncertainty. Inconsistency. Dempster-Shafer theory.	15

III	Nature and Goals of Neural Computing: Comparison with rule-based AI. Overview of network architectures and learning paradigms. Binary Decision Neurons, The McCullough-Pitts model. Single-layer perceptrons and their limitations. The Multilayer Perceptron, The sigmoid output function.Hidden units and feature detectors. Training by error backpropagation. The error surface and local minima. Generalisation, how to avoid overtraining.	15
IV	Introduction to Fuzzy Logic, Classical and Fuzzy Sets: Overview of Classical Sets, Membership Function, Fuzzy rule generation. Operations on Fuzzy Sets: Compliment, Intersections, Unions, Combinations of Operations, Aggregation Operations. Fuzzy Arithmetic: Fuzzy Numbers, Linguistic Variables, Arithmetic Operations on Intervals; Numbers, Lattice of Fuzzy Numbers, Fuzzy Equations. Fuzzy Logic: Classical Logic, Multivalued Logics, Fuzzy Propositions, Fuzzy Qualifiers, Linguistic Hedges.	15
Total Contact Hours		60

Suggested Evaluation Methods

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

Part C-Learning Resources

Reference Books:

1. R. Beale, T. Jackson, " *Neural Computing-an introduction*", CRC Press, 1990.
2. Stuart Russell and Peter Norvig, "*Artificial Intelligence: A Modern Approach*", Prentice-Hall, 4th edition, 2020.
3. George J. Klir and Bo Yuan, "*Fuzzy Sets and Fuzzy Logic: Theory and Applications*" Prentice Hall, 1996.
4. M. Ganesh , "*Introduction To Fuzzy Sets And Fuzzy Logic*", PHI Learning, 2006.

PC-3 PRACTICAL-3 (Based on CC-5 & CC-6)

With effect from Session: 2024-25

Part A - Introduction

Name of the Programme	PGDCA
Semester	2nd
Name of the Course	Practical-3
Course Code	PGD24-CAP-205
Course Type	PC-3
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:	CLO1: Solve practical problems related to theory courses undertaken in the CC-5 and CC-6 from application point of view. CLO 2: Know how to use the server-side web technologies. CLO 3: To implement various languages used for server side web development. CLO 4: To know about programming skills in Java.

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 write and execute 2 questions from the Part-B.	120

Part-A

Node JS:

- Express framework to create web applications: Configuring Routes, Working with Express.
- How to serve Static HTML pages to the browser, and serving other file formats and restricting certain files.
- Database Connectivity: MySQL Database, Creating Connection

Dot NET and C#:

- Polymorphism, Operator Overloading, Interfaces, Delegates and Events.
- Type conversion. C# Using Libraries: Namespace- System, Input-Output, Multi-Threading, Networking and sockets, Managing Console I/O Operations, Windows Forms, Error Handling.
- .Net Assemblies and Attribute: .Net Assemblies features and structure, private and share assemblies, Built-In attribute and custom attribute. Introduction about generic. Frameworks and Libraries:

PHP:

- Function, Call by value, Call by reference, Recursive function, Strings, Arrays, Index based and Associative array.
- Working with file and Directories, State management , Regular expression, Pattern matching.
- Abstract method and class Interface, Final Exception Handling Understanding Exception and error Try, catch, throw.

Ruby:

3) Creating Objects, Member Functions, Variables, Constants and Literals, Operators
4) Conditional Control, Loop Control, Methods, Blocks
5) Modules and Mixins, Strings, Arrays, Hashes, Ranges, Iterators
6) File I/O, Exceptions, DBI, Web Applications.

Java:

- Packages and Exceptions: Java API packages, Creating and using packages, static import, Exceptions handling, Types of Exceptions, multiple catch statements, 'throw' and 'throws', using 'finally' statement, Creating your own exception.
- Applet Programming: How Applets differs from Java Application, Applet Life Cycle, APPLET Tag, Running an Applet, Passing Parameters to Applet.
- Event Handling: Mechanism, The Delegation Event Model, Event Classes, Event Listener Interfaces, Adapter and inner classes.
- GUI Programming: Working with Frame Window, Graphics and Text, AWT Controls and classes. Layout Managers, working with Menus.

Part-B

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

22) Create and handle a custom event using the EventEmitter class
23) Using Express framework, Serve Static HTML and Restrict Access to Certain Files.
24) Connect to a MySQL database and fetch data in Node JS proram.
25) Web Server Handling Requests and Working with Files and Buffers in Node JS.
26) Write a program in C# for Multiple Inheritance using interface.
27) The Car and Motorcycle classes inherit from the base Vehicle class, which defines a StartEngine() method. The StartEngine() method is overridden in the Car and Motorcycle classes to provide specific behavior for each type of vehicle. Use Polymorphism, when the StartEngine() method is called on both Car and Motorcycle objects stored in the Vehicle array.
28) Create a simple Windows Form and handling events. On clicking the "Click Me!", button will display a message box saying "Hello from Windows Forms!". Use file I/O and console I/O operation with error handling to implement the working of form.
29) Write a Program to Demonstrate Assemblies and Attributes.
30) Write a program to handle simple basic terminology in PHP. i)Variable ii)constant iii)data type iv) operator v) function vi) Expression handling vii) keywords viii)Iterations.
31) Write a program in PHP to handle call by value and call by reference.
32) Write a program to handle regular expression and pattern matching.
33) Write a program to handle all the features of OOPs.
34) Create a Ruby program to manage student grades. The program should allow you to add, remove, and update student grades. It should also calculate the average grade and display the highest and lowest grades.
35) Develop a Ruby program to simulate a shopping cart. Users should be able to add items to the cart, remove items, and view the total cost. Include features like discounts and taxes.
36) Create a Ruby program to manage a library's book inventory. The program should allow you to add, remove, and search for books. It should also track which books are checked out and by whom..
37) Develop a Ruby program for a number guessing game. The program should randomly select a number between 1 and 100, and the user should try to guess the number. Provide hints if the guess is too high or too low.
38) 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.
39) Define a Java class representing a Student with private instance variables and public getter and setter methods.
40) Develop a Java program to implement inheritance by creating a base class Animal and derived classes like Dog and Cat.
41) Write a Java program to demonstrate method overriding by implementing a base class Shape and derived classes like Circle and Rectangle.

Suggested Evaluation Methods

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

Part C-Learning Resources

Recommended Books/e-resources/LMS:

1) Mario Casciario (2020). *Node.JS Design Patterns*. Packt Publishers.
2) David Harron (2020). *Node.JS Web Development*. Packt Publishers.
3) Vikram Vaswani (2017). *PHP – A Beginner's Guide*. McGraw Hill Education (Indian Edition)
4) Mark J. Price (2023). *C# 12 and .NET 8*. Packt Publishers.
5) Noel Rappin, Dave Thomas (2024). *Programming Ruby 3.3*. Pragmatic Bookshelf.
6) E. Balagurusamy, "*Programming with Java :A Primer*", McGraw Hill, 3rd edition.
7) Herbert Schildt, "*Java:The Complete Reference*", McGraw Hill, 7th edition.
8) Bruce Eckel, "*Thinking in Java*", Prentice Hall, 4th Edition.
9) Cay S. Horstmann, Gary Cornell, "*Core Java Volume I—Fundamentals*", Prentice Hall, 9th Edition

PC-4 PRACTICAL-4 (Based on CC-7 & CC-8)

With effect from Session: 2024-25

Part A - Introduction

Name of the Programme	PGDCA
Semester	2nd
Name of the Course	Practical-4
Course Code	PGD24-CAP-206
Course Type	PC-4
Level of the course	400-499
Prerequisite 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 Artificial Intelligence and Data Base Management System.
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 techniques of Artificial Intelligence. CLO 3: Know how to create and manipulate Relational Database. CLO 4: Know how to work with MongoDB.

Credits	Theory	Practical	Total
	0	4	4
Teaching Hours per week	0	8	8
Internal Assessment Marks	0	30	30
Term Exam Marks	0	70	70
Total 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: Heuristic Search Techniques**

3) To implement various Heuristic Search Techniques.

Task 2: Neural Networks

3) To know about creation and functioning of Neural Networks.

Task 3: Fuzzy Logic

3) To know about applications of Fuzzy Logic and Fuzzy Arithmetic

- **Task 4: Relational Data Base Management System**

3) Create and Manipulate Relational Data Base Management System.
4) To understand transaction processing in concurrent environment.

- **Task 5: MongoDB**

3) To know about creation of Database in MongoDB
4) To know handling of NoSQL queries.

Part-B

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

24) Write a program implement graph in python using adjacency matrix & Adjacency list
25) Write a program to implement BFS(breadth first search) for graphs in python.
26) Write a program to implement DFS(depth first search) for graphs in python.
27) Write a program to find solution for TSP(Travelling Salesmen problem).
28) Write a program to implement A* algorithm.
29) Write a program to implement AO* algorithm.
30) Write a program to implement Hill Climbing algorithm.
31) Write a program to implement Eight Puzzle Problem
32) Write a program to implementation for propositional logic using python.
33) Create a simple neural network using python
34) Create a knowledge base that include facts and rule using logical reasoning to answer question about knowledge base world.
35) Write a program to implement operation on Fuzzy set.
36) Write a program to code the Defuzzification, fuzzy output to obtain a crisp result using centroid method.
37) Write a Rule based fuzzy logic program to determine fan speed based on temperature.
38) Write a program to find out intersection on fuzzy set using python.
39) Write a program to find out unions on fuzzy set using python.
40) Create a PL/SQL program to manage a student database. The program should include functionalities for adding new students, updating student information, and deleting records. Also, implement procedures to calculate and display student grades and generate reports based on different parameters (e.g., courses, academic year).
41) Write a PL/SQL program to manage a library's book inventory. The program should include procedures for adding new books, updating book details, and removing books from the inventory. Also, implement a feature to keep track of borrowed books and their due dates.
42) Develop a PL/SQL program to manage employee information. The program should allow adding, updating, and deleting employee records. Implement features to calculate employee salaries, bonuses, and to retrieve specific employee details based on various criteria (e.g., department, job title).
43) Develop a PL/SQL program to generate sales reports for a retail store. The program should include functionalities for adding and updating sales data. Implement procedures to generate daily, weekly, and monthly sales reports, and to analyze sales trends.
44) Develop a MongoDB schema to store and manage product information for an e-commerce platform. Implement functionalities to add new products, update product details, remove products, and perform search operations based on product attributes like category, price range, and brand.
45) Develop a MongoDB schema to manage customer feedback for a retail business. Implement functionalities to add new feedback, update existing feedback, delete feedback, and retrieve feedback based on various parameters like product ID, customer ID, and rating.
46) Design a MongoDB schema to manage events for an event planning company. Implement functionalities to create new events, update event details, delete events, and search for events based on date, location, and type. Track attendees and their registration details.
47) Design a MongoDB schema to manage real estate listings for a property management company. Implement functionalities to add new property listings, update listing details, remove listings, and search for properties based on location, price, and property type. Track inquiries and interactions with potential buyers or renters.

Suggested Evaluation Methods

Internal Assessment: 30	**End Term Examination: 70**

	3	Practicum	70
Practicum	0		
4) Class Participation:	5	Lab record, Viva-Voce, write-up and execution of the programs	
5) Seminar/Demonstration/Viva-voce/Lab records etc.:	1 0		
6) Mid-Term Examination:	1 5		

Part C-Learning Resources

Recommended Books/e-resources/LMS:

1. R. Beale, T. Jackson, " *Neural Computing-an introduction*", CRC Press, 1990.
2. Stuart Russell and Peter Norvig, "*Artificial Intelligence: A Modern Approach*", Prentice-Hall, 4th edition, 2020.
3. George J. Klir and Bo Yuan, "*Fuzzy Sets and Fuzzy Logic: Theory and Applications*" Prentice Hall, 1996.
4. M. Ganesh , "*Introduction To Fuzzy Sets And Fuzzy Logic*", PHI Learning, 2006.
5. Ramez Elmasri, Shamkant B. Navathe, *"Fundamentals of Database Systems"*, Pearson, 6th edition, 2010.
6. Silberschatz Abraham , *"Database System Concept"*, Tata Mc Graw Hill, 7th edition, 2019.
7. C. J Date , *"Introduction to Database Systems*", Pearson Education, 8th edition, 2004.
8. Krishnan Ram and Gehrke , *"Database Management System"*, , Tata Mc Graw Hill, 2003.
9. Byross Ivan , *"Oracle 10 G The Database with HTML Database*", BPB publication,2006.
10. Marko Eleksendric et.al., " Mastering MongoDB 7.0", Fourth Edition, Packt Publishers, 2024

With effect from Session: 2024-25

Part A - Introduction

Name of the Programme	PGDCA
Semester	2nd
Name of the Course	Constitutional, Human and Moral Values, and IPR
Course Code	PGD24-CHM-201
Course Type	CHM
Level of the course (As per Annexure-I	400-499
Pre-requisite for the course (if any)	-
Course Objectives	To help the students appreciate the essential complementarily between 'VALUES' and 'SKILLS' to ensure sustained happiness and prosperity which are the core aspirations of all human beings. To facilitate the development of a Holistic perspective among students towards life and profession as well as towards happiness and prosperity based on a correct understanding of the Human reality and the rest of existence. Such a holistic perspective forms the basis of Universal Human Values and movement towards value based living in a natural way. To highlight plausible implications of such a Holistic understanding in terms of ethical human conduct, trustful and mutually fulfilling human behavior and mutually enriching interaction with Nature. To know about publication ethics and Intellectual Property Rights.
Course Learning Outcomes (CLO) After completing this course, the learner will be able to:	CLO-1. To become a good human being with highest moral values. CLO-2 To understand how to generate and maintain harmony in family, society and at workplace. CLO-3 To get the knowledge of Research Ethics and Scientific Misconduct. CLO-4 To implement publication ethics and know about various guidelines.

Credits	Theory	Practical	Total
Teaching Hours per week	4	0	4
Internal Assessment Marks	4	0	4
End Term Exam Marks	30	0	30
Max. Marks	70	0	70
Examination Time	100	0	100
	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	Right Understanding, Relationship and Physical Facility (Holistic Development and the Role of Education), Understanding Value Education, Self-exploration as the Process for Value Education, Continuous Happiness and Prosperity – the Basic Human Aspirations, Happiness and Prosperity – Current Scenario, Method to Fulfill the Basic Human Aspirations, Understanding Human being as the Co-existence of the Self and the Body, Distinguishing between the Needs of the Self and the Body, The Body as an Instrument of the Self, Understanding Harmony in the Self, Harmony of the Self with the Body, Programme to ensure self-regulation and Health	15

	Harmony in the Family – the Basic Unit of Human Interaction, 'Trust' – the Foundational Value in Relationship, 'Respect' – as the Right Evaluation, Other Feelings, Justice in Human-to-Human Relationship, Understanding Harmony in Society, Vision for the Universal Human Order, Understanding Harmony in the Nature, Interconnectedness, self-regulation and Mutual Fulfillment among the Four Orders of Nature, Realizing Existence as Co-existence at All Levels, The Holistic Perception of Harmony in Existence, Definitiveness of (Ethical) Human Conduct, Competence in Professional Ethics, Strategies for Transition towards Value-based Life and Profession	15
III	Introduction to philosophy: definition, nature and scope, concept, branches. Ethics: definition, moral philosophy, nature of moral judgments and reactions. Ethics with respect to science and research, Intellectual honest and research integrity, Scientific misconducts: falsification, fabrication, and plagiarism. Redundant publications: duplicate and overlapping publications, salami slicing Selective reporting and misrepresentation of data	15
IV	Publication ethics: definition, introduction and importance. Best practices/standards setting initiatives and guidelines: COPE, WAME, etc. Conflicts of interest Publication misconduct: definition, concept, problems that lead to unethical behavior and vice versa, Violation of publication ethics, authorship and contributorship Identification of publication misconduct, complaints and appeals, Predatory publishers and journals	15
Total Contact Hours		60

Suggested Evaluation Methods

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

Part C-Learning Resources

Reference Books:

6) JeevanVidya: EkParichaya, A Nagaraj, JeevanVidyaPrakashan, Amarkantak, 1999.
7) Human Values, A.N. Tripathi, New Age Intl. Publishers, New Delhi, 2004.
7) MuralidharK et. al., *Ethics in Science Education, Research and Governance,* Indian National Science Academy, 2019
8) Huma Praveen and NayeemShowkat, *Research Ethics,* e-PG Pathshala, 2017
9) MacIntye, Alasdair, A Short History of Ethics, London, 1967
10) P. Chaddah, Ethics in Competitive Research : Do not get scooped; do not get plagiarized, ISBN: 978-9387480865 (Self Published)