



**MIT ART, DESIGN AND TECHNOLOGY
UNIVERSITY, PUNE**

**MIT SCHOOL OF ENGINEERING
PUNE**

STRUCTURE & SYLLABUS

FOR

**Bachelor of Technology
Computer Science & Engineering
(Software Product Engineering)
Powered By Kalvium**

UNDER FACULTY OF TECHNOLOGY

**4 Year Undergraduate Course approved by BoS &
AC**

(163 CREDITS)

**Department of Computer Science & Engineering
(w.e.f. 2023-24)**

Publisher's Note

MIT Art, Design and Technology University (MITADT), Pune is established under the MIT Art, Design and Technology University Act, 2015 (Mah Act No. XXXIX of 2015). MIT ADT has a great pleasure in publishing the syllabus for **B. Tech Computer Science and Engineering (Software Product Engineering)** under the **Faculty of Engineering**.

On behalf of MIT ADT University, we thank all the experts of various departmental Board of Studies (BoS), who have contributed in designing of syllabus for all branches of Engineering. The syllabus content is designed to incorporate the industry requirement with great emphasis on project based and e-learning. Some of the content delivery and effective teaching methods suggested for student learning are flipped classroom, projects design, solving and implementing real time case studies through innovative practices to improve student learning.

We are thankful to Academic Council to approve the syllabus through debate and discussion on the suggestions giving by BoS. We are also grateful to Board of Management to give their necessary consent to the syllabus and its execution requirement. Finally, we appreciate all people involved in framing and approval of syllabus with their keen interest and whole-hearted co-operation in bringing out this publication.

We are confident that the approved syllabus is most appropriate to provide value based education along with career development skills like industry professionals, pursue higher education/research or to become an entrepreneur.

Head of Department
Dept. of Computer Science and Engineering

Board of Studies Chairman
Dept. of Computer Science and Engineering

Dean
School of Engineering

Registrar,
MIT ADT University,
Loni Kalbhor, Pune-412201



MIT-ADT UNIVERSITY

PUNE, INDIA

A leap towards World Class Education



VISION

MIT Art, Design and Technology University aspires to be the University of Eminence by amalgamating Art, Design, Science and Technology. The University aims to have a transformative impact on society through holistic education, multidisciplinary research ethos, innovation and entrepreneurial culture.



MISSION

- The Mission of MIT Art, Design and Technology University is to provide impetus to faculty, learners, and staff by developing their innate intellectual capabilities, creative abilities and entrepreneurial mind-set for the socio-economic development of the nation.
- We empower learners to become adaptive and agile global professionals through unique specialized programs building academia-industrial partnership.
- We nurture learners to be intellectually curious, technologically equipped, mentally sound, physically fit, spiritually elevated, socio-culturally sensitive, environmentally conscious through continuous holistic education for the ever- evolving world.
- We provide technology-enabled learner-driven curriculum, value added courses, simulated learning environments, state-of-the-art infrastructure and opportunities for community engagement.

School of Engineering



VISION

The School of Engineering aims to achieve **transformative impact** through **academic excellence** and innovation for **socio-sustainable development**.



MISSION

- To disseminate knowledge in computing and multidisciplinary domains with specialized and need-based curriculum, innovating teaching pedagogy and state-of- art infrastructure.
- To develop a culture for high-impact research, innovation and entrepreneurship by creating knowledge through design thinking and project- based learning.
- To imbibe value-based and holistic education for next-generation professionals toward societal development and nation-building.
- To promote conducive work-culture for capacity building of the employees.

Department of Computer Science & Engineering



VISION

The Department of Computer Science Engineering aims to impart progressive education with a state-of-the-art curriculum and generate socially sensitized engineers, innovators, and entrepreneurs for sustainable development.



MISSION

M-1: To pioneer the education in Computer Science and Engineering with a dynamic and industry-ready curriculum.

M-2: To evolve as a Centre of Excellence that contributes significantly to multi-disciplinary research, innovation and entrepreneurship in collaboration with industry and academia.

M-3: To encourage students and faculties by engaging in multi-disciplinary problem-solving and creating sustainable solutions for social well-being.

M-4: To nurture students with ethics and values through holistic education for developing responsible leadership.

Program Outcome (PO's)

Engineering Graduates will be able to:

1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. Individual and teamwork: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. Lifelong learning: Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO1	To apply the acquired knowledge of Computer Science & Engineering to build sustainable solutions to real-world problems in society.
PEO2	Attain the ability to adapt quickly to new environments and technologies, assimilate new information, and work in multi-disciplinary areas with a strong focus on higher studies, innovation and entrepreneurship.
PEO3	To practice collaborative learning and team spirit as professionals with project-based learning and technical activities.
PEO4	To be able to effectively present, communicate and exhibit professionalism with ethical values and empathy for needs of society.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO-1	Apply fundamental knowledge in Computer Science and Engineering with advanced learning in specialized domains of Data Science, Artificial Intelligence, Networking, Security and high-performance computing using standard practices in Engineering and Sciences.
POS-2	Apply standard industry-based practices and techniques to provide solutions to real world problems.

**BTech Computer Science and Engineering Curriculum 2023 Course
Specialization: Software Product Engineering****Semester Credit Mapping**

Sr. No.	Name of the Semester	Credits	Number of Hrs.
1.	Semester 1	25	29
2.	Semester 2	21	24
3.	Semester 3	24	34
4.	Semester 4	24	34
5.	Semester 5	26	35
6.	Semester 6	22	34
7.	Semester 7	22	34
8.	Semester 8	20	38
	Total	184	262

BTech Computer Science and Engineering Curriculum 2023 Course Specialization: Software Product Engineering

FIRST YEAR ENGINEERING SCHEME Semester I

Sr. No.	Course Code	Course	L	T	P	Credits	Total Marks
1	23CSE1111	Fundamentals of Discrete Mathematics	3	1	0	4	100
2	23CSE1112	Introduction to Web Programming - Front end development	3	0	0	3	100
3	23CSE1113	Critical Thinking	3	0	0	3	100
4	23CSE1114	Breadth of Computer Science -1	3	0	0	3	100
5	23CSE1115	Problem Solving using Programming	2	0	2	3	100
6	23CSE1116	Professional Skills for the Workplace	0	1	0	1	100
7	23CSE1117	Design for Developers	0	0	2	1	100
8	23SHD2003	Professional English Communication for Engineers	2	1	0	2	50
9	23SHD1005	Health Practices-I	2	1	0	2	50
10	23SHD1025	Drawing & Painting-1	3	3	0	3	100
		Total	19	2	8	25	900

CA = Continuous Assessment, FE= Final Examination, **Final Lab exam will be conducted with viva-voce of the respective practical (40 exam +10 viva = 50)

Coding for course/ subject: 23CSE1111, Where; 23 = Year of BOS, CSE=Name of Department, 1= Semester No.,

111 to N = Sequence No of Subject. For, SE to BE & also PG follow the above scheme of regulation.

BTech Computer Science and Engineering Curriculum 2023 Course Specialization: Software Product Engineering

FIRST YEAR ENGINEERING SCHEME Semester II

Sr. No.	Course Code	Course	L	T	P	Credits	Total Marks
1	23CSE2111	The Breadth of Computer Science -2	3	0	0	3	100
2	23CSE2112	Full Stack Web Development	3	0	0	3	100
3	23CSE2113	Backend Web Development	2	0	0	2	100
4	23CSE2114	Fundamentals of Databases	3	0	0	3	100
5	23CSE2115	Learning How to learn	2	0	0	2	100
6	23CSE2116	Databases Laboratory	0	0	2	0	100
7	23CSE2117	Full Stack Web Development Laboratory	0	0	2	0	100
8	23SVS1001	Indian Knowledge System for Engineers	1	0	0	1	100
9	23SHD1074	Health Practices-II	1	0	2	1	100
10	23CSE2111	The Breadth of Computer Science -2	3	0	0	3	100
		Total	18	0	06	21	1000

CA = Continuous Assessment, FE= Final Examination, **Final Lab exam will be conducted with viva-voce of the respective practical (40 exam +10 viva = 50)

Coding for course/ subject: 23CSE1111, Where; 23 = Year of BOS, CSE=Name of Department, 2= Semester No.,

111 to N = Sequence No of Subject. For, SE to BE & also PG follow the above scheme of regulation.

**BTech Computer Science and Engineering Curriculum 2023 Course
Specialization: Software Product Engineering**

SECOND YEAR ENGINEERING SCHEME

SEMESTER III

Sr. No.	Course Code	Course Name	Hours/week				Credits	Total Marks
			L	T	P			
1	23CSE3111	Economics, Politics and Rural Society Development	2	1	0	3	100	
2	23CSE3112	Mathematical Thinking 101	3	1	0	4	100	
3	23CSE3113	Databases Management Systems	4	0	0	4	100	
4	23CSE3114	Object Oriented Programming	3	0	0	3	100	
5	23CSE3115	Object Oriented Programming - Lab	0	0	2	1	100	
6	23CSE3116	Database Management Systems - Lab	0	0	2	1	100	
7	23CSE3117	Integrated Work - 1	0	0	16	8	100	
Total			12	2	20	24	700	

CA = Continuous Assessment, FE= Final Examination, **Final Lab exam will be conducted with viva-voce of the respective practical (40 exam +10 viva = 50)

Coding for course/ subject: 23CSE1111, Where; 23 = Year of BOS, CSE=Name of Department, 3= Semester No.,

111 to N = Sequence No of Subject. For, SE to BE & also PG follow the above scheme of regulation.

**BTech Computer Science and Engineering Curriculum 2023 Course
Specialization: Software Product Engineering**

SECOND YEAR ENGINEERING SCHEME

Semester IV

Sr. No.	Course Code	Course Name	Hours/week				Maximum Marks
			L	T	P	Credits	Total
1	23CSE4111	How Human Languages work	3	1	0	4	100
2	23CSE4112	Tools and Techniques for Creative Thinking	3	1	0	4	100
3	23CSE4113	Operating Systems	3	0	0	3	100
4	23CSE4114	Data Structure and Algorithms	3	0	0	3	100
5	23CSE4115	Operating Systems - Lab	0	0	2	1	100
6	23CSE4116	Data Structures and Algorithms - Lab	0	0	2	1	100
7	23CSE4117	Integrated Work - II	0	0	16	8	100
Total			12	2	20	24	700

CA = Continuous Assessment, FE= Final Examination, **Final Lab exam will be conducted with viva-voce of the respective practical (40 exam +10 viva = 50)

Coding for course/ subject: 23CSE1111, Where; 23 = Year of BOS, CSE=Name of Department, 4= Semester No.,

111 to N = Sequence No of Subject. For, SE to BE & also PG follow the above scheme of regulation.

BTech Computer Science and Engineering Curriculum 2023 Course Specialization: Software Product Engineering

THIRD YEAR ENGINEERING SCHEME Semester V

Sr. No.	Course Code	Course Name	Hours/week				Maximum Marks
			Lecture	Tutorial	Practical	Credits	
1	23CSE5111	English LSRW	3	0	0	3	100
2	23CSE5112	Principles of Science	4	0	0	4	100
3	23CSE5113	Computer Organization and Architecture	4	0	0	4	100
4	23CSE5114	Formal Language and Automata Theory	4	0	0	4	100
5	23CSE5115	Design and Analysis of Algorithms	2	0	0	2	100
6	23CSE5116	Design and Analysis of Algorithms - Lab	0	0	2	1	100
7	23CSE5117	Integrated Work - III	0	0	16	8	100
Total			17	0	18	26	700

CA = Continuous Assessment, FE= Final Examination, **Final Lab exam will be conducted with viva-voce of the respective practical (40 exam +10 viva = 50)

Coding for course/ subject: 23CSE1111, Where; 23 = Year of BOS,CSE=Name of Department, 5= Semester No.,

111 to N = Sequence No of Subject. For, SE to BE & also PG follow the above scheme of regulation.

BTech Computer Science and Engineering Curriculum 2023 Course Specialization: Software Product Engineering

THIRD YEAR ENGINEERING SCHEME Semester VI

Sr. No.	Course Code	Course Name	Hours/week				Maximum Marks
			Lecture	Tutorial	Practical	Credits	
1	23CSE6111	Discovering Self	2	2	0	3	100
2	23CSE6112	Fundamentals of Business Management	2	2	0	3	100
3	23CSE6113	Compiler Design	3	0	0	3	100
4	23CSE6114	Computer Networks	3	0	0	3	100
5	23CSE6115	Computer Networks - Lab	0	0	2	1	100
6	23CSE6116	Compiler Design - Lab	0	0	2	1	100
7	23CSE6117	Integrated Work - IV	0	0	16	8	100
Total			10	4	20	22	700

CA = Continuous Assessment, FE= Final Examination, **Final Lab exam will be conducted with viva-voce of the respective practical (40 exam +10 viva = 50)

Coding for course/ subject: 23CSE1111, Where; 23 = Year of BOS, CSE=Name of Department, 6= Semester No.,

111 to N = Sequence No of Subject. For, SE to BE & also PG follow the above scheme of regulation.

BTech Computer Science and Engineering Curriculum 2023 Course Specialization: Software Product Engineering

FINAL YEAR ENGINEERING SCHEME Semester VII

Sr. No.	Course Code	Course Name	Hours/week				Total Marks
			Lecture	Tutorial	Practical	Credits	
1	23CSE7111	Introduction to Philosophy	2	2	0	3	100
2	23CSE7112	Academic Elective 1	3	0	2	4	100
3	23CSE7113	Academic Elective 2	3	0	2	4	100
4	23CSE7114	Foundation Elective	2	2	0	3	100
5	23CSE7115	Integrated Work - V	0	0	16	8	100
	Total		10	4	20	22	500

CA = Continuous Assessment, FE= Final Examination, **Final Lab exam will be conducted with viva-voce of the respective practical (40 exam +10 viva = 50)

Coding for course/ subject: 23CSE1111, Where; 23 = Year of BOS, CSE=Name of Department, 7= Semester No.,

111 to N = Sequence No of Subject. For, SE to BE & also PG follow the above scheme of regulation.

BTech Computer Science and Engineering Curriculum 2023 Course Specialization: Software Product Engineering

FINAL YEAR ENGINEERING SCHEME Semester VIII

Sr. No.	Course Code	Course Name	Hours/week				Maximum Marks
			Lecture	Tutorial	Practical	Credits	Total
1	23CSE8111	Skilling Elective	2	0	4	4	150
2	23CSE8112	Integrated Work - VI	0	0	32	16	200
	Total		2	0	36	20	300

****Final Lab exam will be conducted with viva-voce of the respective practical**

CA = Continuous Assessment, FE= Final Examination, **Final Lab exam will be conducted with viva-voce of the respective practical (40 exam +10 viva = 50)

Coding for course/ subject: 23CSE1111, Where; 23 = Year of BOS, CSE=Name of Department, 8= Semester No.,

111 to N = Sequence No of Subject. For, SE to BE & also PG follow the above scheme of regulation.

SEMESTER I

Course Name Fundamentals of Discrete Mathematics			
Course Code:	23CSE1111	Course Credits:	4
Teaching Hours / Week (L: T: P):	3:1:0	CA Marks:	50
Total Number of Teaching Hours:	45	END-SEM Marks:	50
<p>Course Objectives: (3 to 5)</p> <p>The main objectives of the course are to:</p> <ol style="list-style-type: none"> 1. Conceptual Understanding: Develop a solid grasp of core discrete mathematics concepts such as logic, sets, functions, and relations. 2. Problem-Solving Skills: Apply discrete mathematics principles to solve problems in computer science and related fields. 3. Analytical Thinking: Analyze algorithms, data structures, and mathematical structures for correctness, efficiency, and suitability. 4. Real-World Application: Utilize discrete mathematics to model and solve real-world problems, and effectively communicate solutions both orally and in writing. 			
<p>Course Outcomes:</p> <p>On completion of the course, students will be able to:</p> <ol style="list-style-type: none"> 1. To explain the fundamental concepts and principles of discrete mathematics, including logic, sets, functions, and relations. 2. To apply discrete mathematics concepts to solve problems in computer science. 3. To analyze and evaluate the correctness and efficiency of algorithms, and the validity and soundness of logical arguments. 4. To synthesize discrete mathematics concepts to create mathematical models for real-world problems, such as scheduling and network optimization. 5. To evaluate the strengths and limitations of various discrete mathematics techniques, and make informed decisions about which approach to use in a given context. 			
Unit I	Title - PROOFS		(07 Hours)
Proofs: Introduction and Proofs, Induction, Strong Induction, Number Theory			
Unit II	Title - STRUCTURES		(11 Hours)
STRUCTURE: Graph theory and Colouring, Matching Problems, Minimum Spanning Tree, Communication Networks, Directed graphs, Relations and partial orders, State machines			
Unit III	Title-COUNTING		(07 Hours)

Counting – Part 1: Sums, asymptotics, Divide and Conquer Recurrences, Linear Recurrences
Counting - Part 2 Counting Rules, Generating functions, Infinite sets

Unit IV	Title- PROBABILITY 1	(12 Hours)
Introduction to Probability, Conditional Probability, Independence, Random variables		
Unit V	Title- PROBABILITY 2	(08 Hours)
Expectations, Deviations, Random Walks		
Learning Resources		
Text Books:		
1. Mathematics for Computer Science; Eric Lehman, F Thomson Leighton, Albert R Meyer; 12th Media Services (5 June 2017)		
Reference Books:		
1. Discrete Mathematics and Its Application; Kenneth H Rosen & Dr Kamala Krithivasan; McGraw Hill; 8th edition		
2. A Textbook on Discrete Mathematics; CV Sastry and Rakesh Nayak; Wiley (1 October 2020)M.		

Course Articulation Matrix (CO-PO Mapping) [Fundamentals of Discrete Mathematics-23CSE1111]															
COs	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	Relevant to National/Global/Regional
CO-1	3	2	2	--	--	--	--	--	--	--	--	2	2	2	National/Global
CO-2	3	3	2	3	--	--	--	--	--	--	--	--	2	2	National/Global
CO-3		2	3	--	--	--	--	--	--	--	--	--	2	2	National/Global
CO-4	2	2	3	--	--	--	--	--	--	--	--	--	2	2	National/Global
CO-5	3	2	2	--	2	--	--	--	--	--	--	--	2	2	National/Global
3 – HIGH, 2 – MEDIUM, 1 – LOW															

Course Name			
Introduction to Web Programming - Front end development			
Course Code:	23CSE1112	Course Credits:	3
Teaching Hours / Week (L: T: P):	3:0:0	CA Marks:	50
Total Number of Teaching Hours:	45	END-SEM Marks:	50
Course Objectives: (3 to 5)			
The main objectives of the course are to:			
<ol style="list-style-type: none"> 1. Understand front end basics: HTML, CSS, and JavaScript - This point focuses on providing students with a solid understanding of the fundamental building blocks of web development: HTML for structure, CSS for styling, and JavaScript for interactivity. 2. Create interactive web experiences with JavaScript - Here, students learn how to use JavaScript to add dynamic behavior to web pages, manipulate the Document Object Model (DOM), and create engaging user interactions. 3. Design responsive, cross-browser compatible websites - This objective emphasizes teaching students how to design websites that adapt seamlessly to different screen sizes and devices, ensuring a consistent user experience across various browsers and platforms. 4. Introduction to popular frameworks like Bootstrap or React - This point introduces students to widely-used front end frameworks and libraries, such as Bootstrap or React, empowering them to leverage pre-built components and streamline development processes in their projects. 			
Course Outcomes:			
On completion of the course, students will be able to:			
<ol style="list-style-type: none"> 1. To develop proficiency in HTML, CSS, and JavaScript, and apply them to the development of interactive and responsive web pages. 2. To design and implement web pages that are accessible, user-friendly, and optimized for search engines. 3. To create and use reusable code components to improve productivity and maintainability. 4. To demonstrate an understanding of the principles of web design and user experience, and apply them to front-end development. 5. To use debugging tools and techniques to identify and fix errors in web applications. 			
Unit I	Title - HTML, CSS & JS first steps		(07 Hours)
Environment set up, Introduction to HTML, HTML Block Elements, HTML Inline Elements, HTML Forms, Introduction to CSS, CSS Font & Text, CSS Selectors, CSS Inheritance, CSS Colors, Box Model, Flex Box, JS DOM, Introduction to JS, JS Variables, JS Data Types, Basics of JS Operators, Basics of JS Strings, Basics of JS Conditional Statements, Basics of JS Control Statements, Basics of JS Arrays, Basics of JS Functions, Basics of JS Objects			
Unit II	Title - HTML, CSS & JS Deep dive		(11 Hours)

Part 1 - CSS Advanced Selectors, CSS Positioning, Advanced Flexbox, CSS Grids, Responsive Design, JS Operators, JS Strings, JS Conditional Statements, JS Control Statements, JS Arrays & Functions, JS Objects

Part 2 - JS Advanced Functions, JS Nested Data Structures, JS Higher Order Functions, JS Event Handling, Object Oriented JS, JS Closure, JS Storage, CSS Transition, CSS Animation

Unit III	Title-JS the hard parts	(07 Hours)
JS Prototypal Inheritance, JS Async, JS Callbacks, JS Promises, JS APIs, JS Axios, Unit testing in JS, Deployment		
Unit IV	Title- React first steps	(12 Hours)
Environment set up, Introduction to React, Props & State, Components, React App using Babel, Rendering lists of data, JSX, Hooks, Additional Hooks, Event Handling, Component Lifecycle, Class based components, Routing		
Unit V	Title- React deep dive	(08 Hours)
React Forms, Fetching Data from API, Redux, React Redux, Redux Toolkit, React CSS Library, Material-ui, Front end		

Learning Resources

Text Books:

1. Web development: This book includes: Web development for Beginners in HTML + Web design with CSS + Javascript basics for Beginners; Andy Vickler; Ladoo Publishing LLC (24 May 2021)
2. The Road to Learn React: Your Journey to Master Plain Yet Pragmatic React.Js; Robin Wieruch; Zaccheus Entertainment (1 January 2018)

Reference Books:

1. HTML, CSS, and JavaScript All in One; Julie C. Meloni& Jennifer Kyrnin; Pearson Education; Third edition
2. React and React Native: A complete hands-on guide to modern web and mobile development with React.js; Adam Boduch & Roy Derks; Packt Publishing Limited; 3rd edition

CourseArticulationMatrix(CO-POMapping)[Introduction to Web Programming - Front end development -23CSE1112]															
COs	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	Relevant to National/Global/Regional
CO-1	3	2	2	2	2	--	--	--	--	--	--	--	2	2	National/Global
CO-2	--	--	2	3	2	2	--	--	--	--	--	--	2	2	National/Global
CO-3	2	2	2	3	2		--	--	--	--	--	--	2	2	National/Global
CO-4	3	3	--	--	--	--	--	--	--	--	--	--	2	2	National/Global
CO-5	--	2	2		3	--	--	--	--	--	--	--	2	2	National/Global
3 – HIGH, 2 – MEDIUM, 1 – LOW															

Course Name

Critical Thinking			
Course Code:	23CSE1113	Course Credits:	3
Teaching Hours / Week (L: T: P):	3:0:0	CA Marks:	50
Total Number of Teaching Hours:	45	END-SEM Marks:	50
Course Objectives: (3 to 5)			
The main objectives of the course are to:			
<ol style="list-style-type: none"> 1. Analyze Arguments: Develop skills to critically evaluate the validity and soundness of arguments presented in various contexts, including written texts, speeches, and media. 2. Problem-solving: Cultivate the ability to identify, define, and address complex problems through systematic analysis, creativity, and logical reasoning. 3. Evaluate Information: Learn how to discern credible sources, assess evidence, and differentiate between fact and opinion to make informed judgments and decisions. 4. Construct and Defend Arguments: Practice articulating coherent, well-supported arguments while considering alternative perspectives and anticipating counterarguments to strengthen persuasive communication skills. 			
Course Outcomes:			
On completion of the course, students will be able to:			
<ol style="list-style-type: none"> 1. To describe the cognitive biases and heuristics that can affect human reasoning, and explain how they can lead to thinking errors. 2. To recognize and identify common thinking errors and fallacies in everyday situations. 3. To apply critical thinking skills to analyze and evaluate arguments and evidence. 4. To synthesize ideas and perspectives from different sources to develop reasoned and well-supported arguments. 5. To evaluate the reliability and validity of different sources of information and evidence. 			
Unit I	Title -TWO SYSTEMS OF THINKING	(07 Hours)	
Why think critically, The two systems of thinking, The mental power unit, The lazy system, The marvels of priming, Cognitive ease, Norms, surprises and ease, How judgements work, Jumping to conclusions			
Unit II	Title - HEURISTICS	(11 Hours)	
The law of small numbers, Anchoring effect and Conjunction fallacy			
Unit III	Title-BIASES	(07 Hours)	
Availability bias, Representativeness bias, Survivorship bias, Sunk cost fallacy, Confirmation bias, Google effect and other common biases			
Unit IV	Title- CRITICAL THINKING IN ACTION - I	(12 Hours)	

Assignments on identifying biases in the news, creating fake news, writing an unbiased review, alien travel guide

Unit V	Title- CRITICAL THINKING IN ACTION - II	(08 Hours)
---------------	--	-------------------

Facts vs opinion, worst case scenarios, hypothetical scenarios

Learning Resources

Text Books:

1. Thinking, Fast and Slow; Daniel Kahneman; Penguin 2012 edition

Reference Books:

1. Critical Thinking; Jonathan Haber; The MIT Press; Illustrated edition (7 April 2020) C++ Programming Language; Bjarne Stroustrup; Pearson Education; 4th edition

Course Articulation Matrix (CO-PO Mapping) [Critical Thinking -23CSE1113]															
COs	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	Relevant to National/Global/Regional
CO-1	2	2	2	--	--	--	--	--	--	--	--	--	2	2	National/Global
CO-2	--	3	3	--	--	--	--	--	--	--	--	--	2	2	National/Global
CO-3	--	2	3	--	--	--	--	--	--	--	--	--	2	2	National/Global
CO-4	--	--	--	--	--	3	--	--	--	--	--	--	2	2	National/Global
CO-5	--	--	3	--	--	--	--	--	--	--	--	--	2	2	National/Global
3 – HIGH, 2 – MEDIUM, 1 – LOW															

Course Name Breadth of Computer Science -1			
Course Code:	23CSE1114	Course Credits:	3

Teaching Hours / Week (L: T: P):	3:0:0	CA Marks:	50
Total Number of Teaching Hours:	45	END-SEM Marks:	50
Course Objectives: (3 to 5)			
<p>The main objectives of the course are to:</p> <ol style="list-style-type: none"> 1. Understand Computer Architecture Principles: Students will gain a deep understanding of the principles underlying computer architecture, including the design, functionality, and operation of a simple CPU, and the foundational concepts that govern operating systems. 2. Learn System Software Design: Students will explore the theoretical aspects of system software development, focusing on the processes involved in translating high-level language code into machine code through compilers and assemblers. 3. Analyze Computing System Performance: Students will develop skills in the theoretical analysis of software and hardware performance, learning how to identify bottlenecks, propose optimizations, and understand the trade-offs involved in system design. 4. Conceptualize Operating System Design: Through a theoretical lens, students will study the components and mechanisms of operating systems, understanding how they manage resources, processes, and user interaction. 5. Engage in Conceptual Design and Analysis: Students will apply their knowledge in collaborative discussions and projects to conceptualize the design of a complete computer system, focusing on the integration and interaction between hardware and software components. 			
Course Outcomes:			
<p>On completion of the course, students will be able to:</p> <ol style="list-style-type: none"> 1. To explain the basic principles and components of computer systems, including logic gates, Boolean algebra, and machine language. 2. To design and implement simple algorithms and data structures, and understand their efficiency and limitations. 3. To develop programming skills in multiple languages, and understand their relative strengths and weaknesses. 4. To analyze and evaluate software engineering practices and principles, and apply them to the design and implementation of larger projects. 5. To synthesize different concepts and techniques in computer science to build a complete, working computer system from scratch. 			
Unit I	Title - Boolean functions, Gate Logic, Boolean arithmetic and the ALU	(07 Hours)	
Boolean logic and functions synthesis, Logic gates and HDL, Hardware simulation and multi-bit buses, Binary numbers and binary addition, Negative numbers and ALU, Boolean arithmetic and ALU			
Unit II	Title - Memory, Machine language and Computer architecture and Assembly languages	(11 Hours)	

CO-1	3	--	--	--	--	--	--	--	--	--	--	2	2	2	National/Global
CO-2	2	1	2	2	3	--	--	--	--	--	--	--	2	2	National/Global
CO-3	3	2	2	2	3	--	--	--	--	--	--	--	2	2	National/Global
CO-4	--	3	2	2	2	--	--	--	--	--	--	--	2	2	National/Global
CO-5	2	--	--	--	2	--	--	--	--	--	--	2	2	2	National/Global
3 – HIGH, 2 – MEDIUM, 1 - LOW															

Course Name			
Problem solving using Programming			
Course Code:	23CSE1115	Course Credits:	3
Teaching Hours / Week (L: T: P):	2:0:2	CA Marks:	50
Total Number of Teaching Hours:	45	END-SEM Marks:	50
Course Objectives: (3 to 5)			
The main objectives of the course are to:			
<ol style="list-style-type: none"> 1. Algorithmic Thinking: Develop the ability to break down complex problems into smaller, manageable components and devise step-by-step algorithms to solve them efficiently. 2. Computational Thinking: Cultivate skills in problem abstraction, pattern recognition, and algorithmic design to approach real-world problems from a computational perspective. 3. Data Structures and Algorithms: Learn how to select and implement appropriate data structures and algorithms to optimize problem-solving processes and manage large datasets effectively. 4. Debugging and Optimization: Gain proficiency in identifying and fixing errors in code through systematic debugging techniques, as well as optimizing algorithms and code for improved efficiency and performance. 			
Course Outcomes:			
On completion of the course, students will be able to:			
<ol style="list-style-type: none"> 1. To explain fundamental programming concepts, including data types, control structures, and functions. 2. To apply programming constructs to solve simple problems and algorithms in C/C++ and Python. 3. To analyze and evaluate the efficiency and correctness of algorithms and programs. 4. To synthesize programming constructs to develop larger programs that solve complex problems. 5. To evaluate the strengths and weaknesses of different programming constructs and choose appropriate solutions for different problems. 			
Unit I	Title - ALGORITHMS		(07 Hours)
Computational thinking, Decomposition, Abstraction, Pattern recognition, Algorithms, Writing pseudocode and translating it to code, Looping (While and do-while loops), Variables (Scope, lifetime and initialization), Datatypes (Structures, classes, enums)			
Unit II	Title - VARIABLES		(11 Hours)
Review of variables and their uses, including how to declare and initialize variables; Arithmetic operators, Relational operators, Logical operators, Bitwise operators & Assignment operators; Introduction to strings and advanced string manipulation techniques, such as concatenation, substring extraction, searching, and replacing; Introduction to characters including ASCII and Unicode encoding, character classification functions, and character mapping; Typecasting; Using constants to represent fixed values in code; local and global variables, and their uses and limitations; operator precedence, order of evaluation, and short-circuiting			
Unit III	Title-CONTROL		(07 Hours)

CONTROL Basics –If, If else, for loop & while loop; using boolean expressions to control program flow and evaluate conditions; using switch statements to select one of many possible code paths based on a value or condition; advanced topics such as nested loops, loops with multiple variables, and using loops to iterate

CONTROL Advanced - Nested if, Nested if elseif, Else, for loop, while loop in arrays and strings; Exception handling; using recursive functions to solve problems that can be broken down into smaller sub-problems; advanced techniques for using loops, such as using loop counters, loop flags, and sentinel values; using advanced branching techniques such as the ternary operator and conditional expressions

Unit IV	Title-MODULARITY	(12 Hours)
----------------	-------------------------	--------------------

Organizing code into modules, classes, and functions to improve code structure and reusability; Function parameters; Function return value; reviewing recursion and its use in function design and implementation; Libraries and APIs

Unit V	Title- PROGRAM DEVELOPMENT	(08 Hours)
---------------	-----------------------------------	-------------------

Game development using a programming language, debugging, basic game design principles, user interface design, performance optimization

Learning Resources

Text Books:

1. Think Like a Programmer: An Introduction to Creative Problem Solving by V. Anton Spraul, Released August 2012, published by No Starch Press

Reference Books:

1. The C Programming Language; Brian W. Kernighan & Dennis Ritchie; Pearson Education India; 2nd edition
2. Programming in Python 3: A Complete Introduction to the Python Language; Mark Summerfield; Pearson Education; Second edition
3. C++ Programming Language; Bjarne Stroustrup; Pearson Education; 4th edition

Course Articulation Matrix (CO-PO Mapping) [Problem solving using Programming -23CSE1115]															
COs	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	Relevant to National/Global/Regional
CO-1	3	--	--	--	--	--	--	--	--	--	--	--	2	2	National/Global
CO-2	2	3	1	2	2	--	--	--	--	--	--	--	2	2	National/Global
CO-3	2	2	2	2	--	--	--	--	--	--	--	--	2	2	National/Global
CO-4	--	2	3	--	--	--	--	--	--	--	--	--	2	2	National/Global
CO-5	--	2	2	2	2	--	--	--	--	--	--	--	2	2	National/Global
3 – HIGH, 2 – MEDIUM, 1 – LOW															

Course Name			
Professional skills for the workplace			
Course Code:	23CSE1116	Course Credits:	1

Teaching Hours / Week (L: T: P):	0:1:0	CA Marks:	100
Total Number of Teaching Hours:	45	END-SEM Marks: -	NA
Course Objectives: (3 to 5)			
The main objectives of the course are to:			
<ol style="list-style-type: none"> 1. Develop effective communication skills, including verbal, nonverbal, and written communication, to convey ideas clearly and professionally in various workplace contexts. 2. Enhance interpersonal skills to build positive relationships with colleagues, clients, and stakeholders, fostering teamwork, collaboration, and conflict resolution. 3. Acquire time management and organization techniques to prioritize tasks, meet deadlines, and maximize productivity in a fast-paced work environment. 4. Cultivate critical thinking and problem-solving abilities to analyze complex situations, make informed decisions, and adapt to changes in the workplace effectively. 			
Course Outcomes:			
On completion of the course, students will be able to:			
<ol style="list-style-type: none"> 1. To explain the four quadrants of the Human Skills Matrix and how they relate to success in the workplace. 2. To apply critical thinking skills to solve complex problems and make effective decisions. 3. To lead and collaborate effectively with others, including managing teams and facilitating group discussions. 4. To communicate clearly and persuasively in various professional contexts, including written, verbal, and nonverbal communication. 5. To manage time and prioritize tasks effectively, including setting goals and developing strategies for self-motivation and self-improvement. 			
Unit I	Title - Managing ourselves		(07 Hours)
Self-awareness, Adaptability, Managing emotions, Accountability, Professionalism, Taking initiative, Persistence, Planning and organizing, Integrity			
Unit II	Title - Interacting - I		(11 Hours)
Writing good emails & reports, Use of communication tools, Negotiation skills,			
Unit III	Title-Interacting -II		(07 Hours)
Use of project management tools, Curating relationships on social media through professional networking sites, Presentation skills			
Unit IV	Title- Thinking		(12 Hours)
Ethical dilemmas, being an intrapreneur, Building a growth mindset, Systems thinking			
Unit V	Title- Leading		(08 Hours)

Empowering others, Having a strategic vision, Project management, Performance management

Learning Resources

Text Books:

1. COMMUNICATION SKILLS FOR PROFESSIONALS AND STUDENTS: An Occupational Therapist's Perspective; Dr. Amitabh Kishor Dwivedi; Notion Press; 1st edition7)

Reference Books:

1. The Communication Book: 44 Ideas for Better Conversations Every Day; Mikael Krogerus& Roman Tschäppeler; Portfolio Penguin (19 April 2018)

Course Articulation Matrix(CO-PO Mapping)[Professional skills for the workplace -23CSE1116]

COs	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	Relevant to National/Global/Regional
CO-1	3	--	--	--	--	--	--	--	--	--	--	--	2	2	National/Global
CO-2	--	2	3	3	2	--	--	--	--	--	--	--	2	2	National/Global
CO-3	2	--	--	--	--	3	2	--	--	--	--	--	2	2	National/Global
CO-4	2	--	--	--	--	2	3	--	--	--	--	--	2	2	National/Global
CO-5	--	--	--	2	--	2	2	2	--	--	--	--	2	2	National/Global

3 – HIGH, 2 – MEDIUM, 1 – LOW

Course Name Design for Developers			
Course Code:	23CSE1117	Course Credits:	1
Teaching Hours / Week (L: T: P):	0:0:2	CA Marks:	100
Total Number of Teaching Hours:	45	END-SEM Marks: -	NA
<p>Course Objectives: (3 to 5)</p> <p>The main objectives of the course are to:</p> <ol style="list-style-type: none"> 1. Understanding Design Principles: Introduce fundamental design principles such as layout, typography, color theory, and user experience (UX) to equip developers with the knowledge to create visually appealing and user-friendly interfaces. 2. Hands-on Design Tools Proficiency: Provide practical training on popular design tools like Adobe XD, Figma, or Sketch, enabling developers to translate design concepts into tangible prototypes and mockups. 3. Collaborative Design Workflow: Foster collaboration between developers and designers by teaching methodologies like Agile, Design Thinking, or Lean UX, ensuring seamless integration of design and development processes. 4. Responsive Design Implementation: Explore techniques for implementing responsive design, ensuring that applications and websites adapt effectively across various devices and screen sizes, enhancing accessibility and user satisfaction. 			
<p>Course Outcomes:</p> <p>On completion of the course, students will be able to:</p> <ol style="list-style-type: none"> 1. To explain the basic principles of user-centered design, and how they apply to software development. 2. To evaluate and critique the design of existing software applications, and identify areas for improvement in terms of user experience and usability. 3. To apply visual design principles and techniques, such as typography, color theory, and layout, to create effective user interfaces. 4. To conduct usability testing and other evaluation methods to measure the effectiveness and usability of software applications. 5. To synthesize different design concepts and techniques to create well-designed and user-friendly software interfaces. 			
Unit I	Title -How to approach design as a developer	(07 Hours)	
Why design for developers, Getting started with design education, Getting started with Figma, Labs on Figma			
Unit II	Title - UX Design for Developers - I	(11 Hours)	

Understanding design sprints, Understanding the user, Information architecture, Accessibility in design		
Unit III	Title-UX Design for Developers -II	(07 Hours)
How to build good digital products, Psychology in UX design, User research, User journey mapping, Wireframing, Prototyping		
Unit IV	Title- UI Design for Developers -I	(12 Hours)
Visual design principles, Visual hierarchy, Working with text, Layouts and spacing		
Unit V	Title-UI Design for Developers -II	(08 Hours)
Working with colours, Working with images, Creating depth in design, Working with components		
Learning Resources		
Text Books:		
1. About Face; Alan Cooper, Robert Reimann, Christopher Noessel and David Cronin; Wiley Publishing, 2014		
Reference Books:		
1. Hands-on UX design for developers; Elvis Canziba; Packt, 2018		
2. Refactoring; Adam Wathan and Steve Shoger		

Course Articulation Matrix (CO-PO Mapping) [Design for Developers -23CSE1117]															
COs	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	Relevant to National/Global/Regional
CO-1	3	2	2	2	3	--	--	--	--	--	--	--	2	2	National/Global
CO-2	2	2	3	3	2	--	--	--	--	--	--	--	2	2	National/Global
CO-3	2	2	2	3	3	--	--	--	--	--	--	--	2	2	National/Global
CO-4	--	2	2	2	2	--	--	--	--	--	--	--	2	2	National/Global
CO-5	--	--	--	--	3	2	--	--	--	--	--	--	2	2	National/Global
3 – HIGH, 2 – MEDIUM, 1 - LOW															

SEMESTER II

Course Name			
The Breadth of Computer Science -2			
Course Code:	23CSE2111	Course Credits:	3
Teaching Hours / Week (L: T: P):	3:0:0	CA Marks:	50
Total Number of Teaching Hours:	48	END-SEM Marks:	50
<p>Course Objectives: (3 to 5)</p> <p>The main objectives of the course are:</p> <ol style="list-style-type: none"> 1. Understand Computer Architecture Principles: Students will gain a deep understanding of the principles underlying computer architecture, including the design, functionality, and operation of a simple CPU, and the foundational concepts that govern operating systems. 2. Learn System Software Design: Students will explore the theoretical aspects of system software development, focusing on the processes involved in translating high-level language code into machine code through compilers and assemblers. 3. Analyze Computing System Performance: Students will develop skills in the theoretical analysis of software and hardware performance, learning how to identify bottlenecks, propose optimizations, and understand the trade-offs involved in system design. 4. Conceptualize Operating System Design: Through a theoretical lens, students will study the components and mechanisms of operating systems, understanding how they manage resources, processes, and user interaction. 5. Engage in Conceptual Design and Analysis: Students will apply their knowledge in collaborative discussions and projects to conceptualize the design of a complete computer system, focusing on the integration and interaction between hardware and software components. 			
<p>Course Outcomes:</p> <p>On completion of the course, students will be able:</p> <ol style="list-style-type: none"> 1. To design and implement a simple CPU. 2. To write an assembler to translate assembly language code into machine code. 3. To write a compiler to translate high-level language code into assembly language code. 4. To design and implement a simple operating system. 5. To analyze and optimize the performance of software and hardware components. 			
Unit I	High level language	(12 Hours)	
The Jack Language in a nutshell , Object-Based Programming, List Processing, Jack Language, Specification: Syntax and Data Types, Jack Language Specification: Classes and methods, Lab on Classes and methods, Developing Apps using the Jack language and OS			
Unit II	Syntax analysis	(12 Hours)	
A Sample Jack App: Square Dance, Graphics Optimization, Syntax Analysis, Lexical Analysis, Grammars, Parse Trees & Parse Logic, The Jack Grammar, The Jack Analyzer			

Unit III	Code generation	(12 Hours)
Code Generation, Handling Variables, Handling Expressions, Handling Flow of Control, Handling Objects: Low-Level Aspects, Handling Objects: Construction, Handling Objects: Manipulation, Handling Arrays, Standard Mapping Over the Virtual Machine		
Unit IV	Operating system	(12 Hours)
Operating System, Efficiency Matters, Mathematical Operations, Memory Access, Heap Management, Graphics, Line Drawing, Handling Textual Output, Input, String Processing, Array Processing, The Sys Class		
Learning Resources		
Text Books: The Elements of Computing Systems: Building a Modern Computer from First Principles by Noam Nisan and Shimon Schocken, published by MIT Press. Latest edition: 2nd edition (August 2019).		
Reference Books: 1. Computer Architecture: A Quantitative Approach by John L. Hennessy and David A. Patterson, published by Elsevier. Latest edition: 6th edition (June 2021). 2. Compilers: Principles, Techniques, and Tools by Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman, published by Pearson. Latest edition: 2nd edition (September 2006).		

Course Articulation Matrix (CO-PO Mapping) [The Breadth of Computer Science -2-23CSE2111]															
COs	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	Relevant to National/Global/Regional
CO-1	2	--	--	3	3	--	--	--	--	--	--	--	2	2	National/Global
CO-2	2	2	2	2	3	--	2	--	--	--	--	--	2	2	National/Global
CO-3	2	2	2	2	3	--	2	--	--	--	--	--	2	2	National/Global
CO-4	2	--	--	3	3	--	--	--	--	--	--	--	2	2	National/Global
CO-5	--	2	2		2	--	--	--	--	--	--	--	2	2	National/Global
3 – HIGH, 2 – MEDIUM, 1 – LOW															

Course Name Full Stack Web Development			
Course Code:	23CSE2112	Course Credits:	3

Teaching Hours / Week (L: T: P):	3:0:0	CA Marks:	50
Total Number of Teaching Hours:	48	END-SEM Marks:	50
<p>Course Objectives: (3 to 5)</p> <p>The main objectives of the course are to:</p> <ol style="list-style-type: none"> 1. Comprehensive Understanding of Full Stack Technologies: Students will gain a thorough understanding of full stack web development, including the architecture, technologies, and interactions between the front-end, back-end, and databases. 2. Database Design and Management with MongoDB: Students will learn to theoretically design and develop database schemas with MongoDB, understanding data modeling, index creation, and query writing to support web applications. 3. Development of RESTful APIs: Through theoretical exploration, students will learn to create RESTful APIs using Node.js and Express, focusing on handling HTTP methods, middleware integration, and database interactions. 4. Front-end Development with React: Students will understand the principles of developing user interfaces using React, including component-based architecture, state management, and event handling for dynamic web applications. 5. Integration and Project Design: Students will conceptualize the integration of front-end and back-end components to build fully functional web applications, culminating in a theoretical project where they design, develop, and test a unique full stack application based on user requirements. 			
<p>Course Outcomes:</p> <p>On completion of the course, students will be able:</p> <ol style="list-style-type: none"> 1. To describe the architecture and components of a full stack web application, including front-end and back-end technologies and their interactions. 2. To design and develop a database schema using MongoDB, including defining data models, creating indexes, and writing queries. 3. To create RESTful APIs using Node.js and Express, including handling HTTP requests and responses, and interacting with the database. 4. To implement front-end user interfaces using React, including using React components, managing state, and handling user input. 5. To integrate front-end and back-end components to create a fully functional full stack web application, including using asynchronous communication and handling errors and exceptions. 			
Unit I	Environment setup	(12 Hours)	
Creating a project and understanding project structures, Using GitHub for version control and project sharing, Setting up the development environment for Full Stack development, Launching a project.			
Unit II	Building and Deploying the Backend	(12 Hours)	
Making the first API and understanding REST principles, Deploying APIs to local and production environments, Testing APIs with an API client and initial database setup, Writing Mongoose schemas for application entities.			
Unit III	Integrating Database with Backend	(12 Hours)	

Setting up a database and connecting it to the Express server, Modifying database records through APIs, Understanding relations between entities and their implementation in schemas, Adding users in the database and handling validations.

Unit IV	Frontend Development and Integration	(12 Hours)
----------------	---	--------------------

Setting up a simple React application with Vite, Deploying the Front-end and understanding deployment strategies, Designing and using components for CRUD operations in React, Implementing user authentication and session management with cookies.

Learning Resources

Text Books:

Full-Stack React Projects: Learn MERN stack development by building modern web apps using MongoDB, Express, React, and Node.js; ShamaHoque; Packt Publishing Limited; 2nd edition

Reference Books:

1. Beginning MERN Stack: Build and Deploy a Full Stack MongoDB, Express, React, Node.js App; Greg Lim
2. Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node; Vasam Subramanian; Apress; 2nd edition

Course Articulation Matrix (CO-PO Mapping) [Full Stack Web Development - 23CSE2112]															
COs	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	Relevant to National/Global/Regional
CO-1	3	--	--	--	3	--	--	--	--	--	--	--	2	2	National/Global
CO-2	3	--	--	2	3	--	--	--	--	--	--	--	2	2	National/Global
CO-3	2	--	--	--	3	--	--	--	--	--	--	--	2	2	National/Global
CO-4	2	--	--	--	3	--	--	--	--	--	--	--	2	2	National/Global
CO-5	--	2	3	2	3	--	--	--	--	--	--	--	2	2	National/Global
3 – HIGH, 2 – MEDIUM, 1 – LOW															

Course Name			
Backend Web Development			
Course Code:	23CSE2113	Course Credits:	2
Teaching Hours / Week (L: T: P):	2:0:0	CA Marks:	50
Total Number of Teaching Hours:	32	END-SEM Marks:	50

Course Objectives: (3 to 5)

The main objectives of the course are:

1. Master Backend Development Concepts: Understand the fundamental concepts of backend web development, including APIs, databases, and server-side scripting.
2. Understand Application Lifecycle: Gain insights into the development and deployment lifecycle of backend applications, emphasizing RESTful API functionality and database integration.
3. Practical Tool Configuration: Theoretically configure and utilize development tools and environments for backend application creation, including API setup with Express and database connectivity.
4. Codebase Analysis Skills: Develop the ability to analyze and critique backend codebases, identifying key components like routing, middleware, and database schema design.
5. Design and Evaluate Backend Systems: Synthesize knowledge to theoretically design comprehensive backend systems, focusing on security, performance, and advanced features like user authentication and data validation.

Course Outcomes:

On completion of the course, students will be able to:

1. To recall and list the fundamental concepts of backend web development, including the roles of APIs, databases, and server-side scripting.
2. To demonstrate an understanding of the development and deployment lifecycle of backend applications, including the purpose and functionality of RESTful APIs and the significance of database integration.
3. To apply their knowledge to configure and use development tools and environments for creating backend applications, demonstrating the ability to set up a basic API using Express and connect it to a database.
4. To analyze and dissect backend codebases to identify components of API development, including routing, middleware, and error handling, as well as understand the structure and design of database schemas relevant to backend processes.
5. To evaluate the security and performance aspects of backend applications, critiquing and suggesting improvements for database interactions, API security practices (such as authentication and authorization), and data validation methods.

Unit I	Backend essentials	(8 Hours)
Creating the first API and page - an introduction to Express, Deploying APIs: from development to production environments, API testing with clients and initial database integration, Advanced API features: Authentication and security practices.		
Unit II	Database Integration and Schema Design	(8 Hours)
Database setup and connection with Express, CRUD operations through API endpoints, Writing and implementing Mongoose schemas, Relations between entities in a database model.		
Unit III	Security and Data Management	(8 Hours)

Adding users to the database and implementing validation logic, Securely managing passwords and sensitive user information, Secure and insecure practices in handling cookies and sessions, Advanced database management techniques.

Unit IV	Deployment and Testing	(8 Hours)
Deploying backend services to production, Ensuring backend performance and security, Testing strategies for backend applications, Continuous integration and deployment practices.		

Learning Resources

Text Books:

Full-Stack React Projects: Learn MERN stack development by building modern web apps using MongoDB, Express, React, and Node.js; ShamaHoque; Packt Publishing Limited; 2nd edition

Reference Books:

1. Beginning MERN Stack: Build and Deploy a Full Stack MongoDB, Express, React, Node.js App; Greg Lim
2. Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node; Vasam Subramanian; Apress; 2nd edition

Course Articulation Matrix (CO-PO Mapping) [Backend Web Development -23CSE2113]

COs	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	Relevant to National/Global/Regional
CO-1	3	--	--	--	3	--	--	--	--	--	--	--	2	2	National/Global
CO-2	3	2	2	3	3	--	--	--	--	--	--	--	2	2	National/Global
CO-3	2	2	2	3	3	--	--	--	--	--	--	--	2	2	National/Global
CO-4	3	--	2	--	3	--	--	--	--	--	--	--	2	2	National/Global
CO-5	--	2	2	--	3	--	--	--	--	--	--	--	2	2	National/Global

3 – HIGH, 2 – MEDIUM, 1 – LOW

Course Name Fundamentals of Databases			
Course Code:	23CSE2114	Course Credits:	3
Teaching Hours / Week (L: T: P):	3:0:0	CA Marks:	50
Total Number of Teaching Hours:	48	END-SEM Marks:	50

Course Objectives: (3 to 5)

The main objectives of the course are to:

1. Understand NoSQL Fundamentals: Grasp the core principles, types, and use cases of NoSQL databases, distinguishing them from traditional relational databases.
2. Architectural Insights: Gain a comprehensive understanding of the architectural differences between NoSQL and relational databases, and the scenarios in which NoSQL is preferred.
3. Design Theoretical NoSQL Schemas: Apply theoretical knowledge to design NoSQL database schemas for given scenarios, selecting appropriate database types for specific needs.
4. Analyze Database Performance: Assess the strengths, limitations, scalability, and performance of various NoSQL databases, understanding their impact on data models and application requirements.
5. Innovate with NoSQL Solutions: Synthesize course concepts to propose theoretical, innovative solutions using NoSQL databases for addressing complex data challenges.

Course Outcomes:

On completion of the course, students will be able:

1. To identify and recall the core principles of NoSQL databases, including the types of NoSQL databases (document, key-value, wide-column, and graph databases) and their primary use cases.
2. To explain the architectural differences between NoSQL and traditional relational databases, understanding when and why NoSQL databases are chosen over relational databases for certain applications.
3. To apply theoretical knowledge to design NoSQL database schemas for hypothetical scenarios, demonstrating an ability to choose the appropriate NoSQL database type based on specific requirements.
4. To analyze the strengths and limitations of different NoSQL databases, assessing their scalability, performance, and suitability for various data models and application needs.
5. To critically evaluate case studies of NoSQL database implementations, discussing the impact of database design decisions on application performance, scalability, and maintainability.

Unit I	Introduction to databases	(12 Hours)
Setting up databases for web applications, Understanding schema design and data modeling, Implementing relations between entities, Basic database operations and CRUD fundamentals.		
Unit II	Advanced Database Operations	(12 Hours)
Modifying and managing data through APIs, Advanced schema design with Mongoose, Integrating database services with backend applications, Database security and user management.		
Unit III	Performance and Optimization	(12 Hours)
Optimizing database queries for performance, Managing database connections and pools, Data indexing strategies and best practices, Analyzing and improving database performance.		
Unit IV	Security and maintenance	(12 Hours)

Securely managing user data and passwords, Implementing authentication and authorization in databases, Database backup and recovery strategies, Ensuring data integrity and security.

Learning Resources

Text Books:

Full-Stack React Projects: Learn MERN stack development by building modern web apps using MongoDB, Express, React, and Node.js; ShamaHoque; Packt Publishing Limited; 2nd edition

Reference Books:

1. Beginning MERN Stack: Build and Deploy a Full Stack MongoDB, Express, React, Node.js App; Greg Lim
2. Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node; Vasam Subramanian; Apress; 2nd edition

Course Articulation Matrix (CO-PO Mapping) [Fundamentals of Databases -23CSE2114]

COs	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	Relevant to National/Global/Regional
CO-1	3	--	--	--	--	--	--	--	--	--	--	--	2	2	National/Global
CO-2	3	--	2	2	2	--	--	--	--	--	--	--	2	2	National/Global
CO-3	2	3	2	3	3	--	--	--	--	--	--	--	2	2	National/Global
CO-4	--	--	3	--	3	--	--	--	--	--	--	--	2	2	National/Global
CO-5		2	3	--	2	2	2	--	2	--	--	--	2	2	National/Global

3 – HIGH, 2 – MEDIUM, 1 – LOW

Course Name

Learning how to learn

Course Code:	23CSE2115	Course Credits:	2
Teaching Hours / Week (L: T: P):	2:0:0	CA Marks:	50
Total Number of Teaching Hours:	48	END-SEM Marks:	50

Course Objectives: (3 to 5)

The main objectives of the course are to:

1. Grasp Effective Learning Principles: Understand and explain the fundamental principles and processes that underpin effective learning across various subject areas.
2. Enhance Memory and Retention: Employ and apply a range of memory-enhancing techniques, including spaced repetition and visualization, to boost information retention and recall.
3. Develop Metacognitive Skills: Practice and refine metacognitive strategies, such as self-reflection and self-assessment, to continually monitor and enhance one's own learning progress.
4. Address Learning Obstacles: Identify common barriers to learning, including procrastination and distractions, and develop strategies to overcome these challenges.
5. Apply and Evaluate Learning Strategies: Integrate and apply various learning techniques to real-world challenges, and critically evaluate their effectiveness to optimize learning outcomes.

Course Outcomes:

On completion of the course, students will be able to:

1. To explain the key principles and processes involved in effective learning, and how they can be applied to any subject area.
2. To apply a variety of memory techniques, such as spaced repetition and visualization, to improve retention and recall of information.
3. To practice metacognitive strategies, such as self-reflection and self-assessment, to monitor and improve learning progress.
4. To identify and overcome common obstacles to effective learning, such as procrastination and distractions.
5. To synthesize different learning techniques and apply them to real-world learning challenges, such as preparing for exams or learning new skills.

Unit I	How learning works	(12 Hours)
Introduction to Focus and Diffuse Modes, the role of Practice in learning, Introduction to Memory & Sleep in learning, Procrastination on learning, case studies to end the unit (on learning languages, creativity, problem solving and ,much more – how learning happens for these skills)		
Unit II	Chunking	(12 hours)
Introduction to Chunking, how to form a Chunk, Illusions of Competence, the pitfalls of overlearning & choking		
Unit III	Procrastination, Memory and Learning	(12 hours)
Avoiding rut thinking, strategies and techniques to tackle procrastination while learning, Process vs. Product, Deep-dive into memory, Long-term memory, Being a life-long learner, The Memory Palace technique		

Unit IV	Being a better learner	(12 hours)
Creating Visual metaphors, creating analogies for learning, Importance of checklists, Learning vs. Prepping for tests, Case studies		
Learning Resources		
Text Books: <ol style="list-style-type: none"> 1. Learning How to Learn: How to Succeed in School Without Spending All Your Time Studying; A Guide for Kids and Teens; Barbara Oakley and Terrence Sejnowski; TarcherPerigree 2018 2. Make it Stick: The Science of Successful Learning; Peter C Brown, Henry L. Roediger III & Mark A. McDaniel; Harvard University Press; 1st edition 		
Reference Books: <ol style="list-style-type: none"> 1. Mindshift: Break Through Obstacles to Learning and Discover Your Hidden Potential; Barbara Oakley, Penguin, 2017 2. How we learn: The surprising truth about when, where and why it happens; Benedict Carey, Random House, 2014 3. Ultralearning: Accelerate Your Career, Master Hard Skills and Outsmart the Competition; Scott H Young, Harper Collins, 2019 		

Course Articulation Matrix (CO-PO Mapping) [Learning how to learn -23CSE2115]															
COs	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	Relevant to National/Global/Regional
CO-1	2	--	--	--	--	--	2	--	--	--	--	--	2	2	National/Global
CO-2	2	2	--	--	3	--	--	--	--	--	--	--	2	2	National/Global
CO-3	2	2	2	--	--	--	--	--	--	--	--	--	2	2	National/Global
CO-4	--	3	--	--	--	--	--	--	--	--	--	--	2	2	National/Global
CO-5	--	2	2	--	3	--	--	--	--	--	--	--	2	2	National/Global
3 – HIGH, 2 – MEDIUM, 1 - LOW															

Course Name Databases Laboratory			
Course Code:	23CSE2116	Course Credits:	1
Teaching Hours / Week (L: T: P):	0:0:2	CA Marks:	100
Total Number of Teaching Hours:	32	END-SEM Marks:	NA

Course Objectives: (3 to 5)

The main objectives of the course are to:

1. Master NoSQL CRUD Operations: Demonstrate proficiency in performing basic database operations—create, read, update, delete (CRUD)—across various NoSQL systems.
2. Understand NoSQL Database Types: Clearly explain the differences, operational uses, and advantages of various NoSQL databases, including document, key-value, wide-column, and graph databases.
3. Design and Implement NoSQL Schemas: Apply theoretical knowledge to practically design and implement efficient NoSQL database schemas, tailored to meet specific application requirements.
4. Optimize NoSQL Queries: Analyze and optimize NoSQL database queries for improved efficiency and speed, with a focus on the performance impacts of schema design.
5. Evaluate Database Performance: Conduct practical experiments to evaluate the scalability and resilience of NoSQL databases, and develop NoSQL-based solutions for realistic, complex scenarios, integrating these databases into full-stack applications.

Course Outcomes:

On completion of the course, students will be able:

1. To recall and demonstrate basic operations in NoSQL databases, such as creating, reading, updating, and deleting data (CRUD operations) in various NoSQL systems
2. To explain the operational differences and use cases between different types of NoSQL databases (document, key-value, wide-column, and graph databases)
3. To apply their knowledge to design and implement NoSQL database schemas based on given requirements, utilizing the appropriate database type for specific application needs
4. To analyze and optimize NoSQL database queries for efficiency and speed, considering the impact of their schema designs on performance
5. To evaluate the scalability and resilience of NoSQL databases through practical experiments, assessing how different designs affect the database's ability to scale and maintain performance under load

Unit I	Introduction to NoSQL Databases	(8 Hours)
Overview of NoSQL vs. Relational Databases, Types of NoSQL Databases: Document, Key-Value, Wide-Column, and Graph, Setting up a NoSQL Database Environment, Basic CRUD Operations in Document and Key-Value Stores		
Unit II	Advanced NoSQL Database Operations	(8 hours)
Designing NoSQL Database Schemas, Advanced Querying Techniques, Indexing and Performance Optimization, Handling Relationships and Embedding in Document Stores		
Unit III	Scalability and Performance	(8 hours)
Scaling NoSQL Databases: Sharding and Replication Strategies, Performance Benchmarking and Monitoring, Implementing Caching Mechanisms, Data Consistency and Partition Tolerance in Distributed Systems		
Unit IV	Integrating NoSQL Databases with Applications	(8 hours)

Connecting NoSQL Databases to Backend Applications, Session Management and User Authentication, Implementing Complex Queries and Aggregations, Project: Building a Full-Stack Application with a NoSQL Backend

Learning Resources

Text Books:

Full-Stack React Projects: Learn MERN stack development by building modern web apps using MongoDB, Express, React, and Node.js; ShamaHoque; Packt Publishing Limited; 2nd edition

Reference Books:

1. Beginning MERN Stack: Build and Deploy a Full Stack MongoDB, Express, React, Node.js App; Greg Lim
2. Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node; Vasanth Subramanian; Apress; 2nd edition

Course Articulation Matrix (CO-POMapping) [Databases Laboratory -23CSE2116]

COs	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	Relevant to National/Global/Regional
CO-1	3	--	--	--	2	--	--	--	--	--	--	--	2	2	National/Global
CO-2	3	--	2	2	2	--	--	--	--	--	--	--	2	2	National/Global
CO-3	2	3	2	3	3	--	--	--	--	--	--	--	2	2	National/Global
CO-4	--	--	3	--	3	--	--	--	--	--	--	--	2	2	National/Global
CO-5	--	2	3	--	3	2	2	--	2	--	--	--	2	2	National/Global

3 – HIGH, 2 – MEDIUM, 1 – LOW

Course Name			
Full Stack Web Development Laboratory			
Course Code:	23CSE2117	Course Credits:	1
Teaching Hours / Week (L: T: P):	0:0:2	CA Marks:	100
Total Number of Teaching Hours:	32	END-SEM Marks:	NA
Course Objectives: (3 to 5)			
The main objectives of the course are to:			
<ol style="list-style-type: none"> 1. Master Frontend and Backend Basics: Demonstrate the application of fundamental web technologies—HTML, CSS, JavaScript for the frontend, and Node.js with Express for the backend services. 2. Understand Web Application Design Principles: Clearly explain and apply the principles of responsive web design, RESTful APIs, and the integration process of frontend and backend components in web applications. 3. Develop Interactive Web Applications: Utilize knowledge in both frontend and backend development to create simple yet interactive web applications, showcasing client-side and server-side functionality. 4. Architect Web Applications: Analyze requirements for web applications to determine the most effective architecture, making informed decisions on database selection, API design, and choosing frontend frameworks. 5. Implement Advanced Full-Stack Projects: Design and develop complex full-stack web applications that address real-world challenges, proving the ability to integrate frontend, backend, and database technologies into a seamless and efficient system. 			
Course Outcomes:			
On completion of the course, students will be able:			
<ol style="list-style-type: none"> 1. To recall and demonstrate the use of basic HTML, CSS, and JavaScript for frontend development, along with Node.js and Express for backend services. 2. To explain the principles of responsive web design, RESTful API concepts, and the process of integrating frontend and backend components. 3. To apply their knowledge to build simple interactive web applications that include both client-side and server-side functionalities. 4. To analyze web application requirements to determine the most effective architecture, including the choice of database, API design, and frontend framework. 5. To evaluate the performance, scalability, and security of web applications, implementing best practices to optimize application functionality. 			
Unit I	Setting up the Full Stack Environment	(8 Hours)	
Introduction to Full Stack Development, Setting up the Development Environment (IDEs, Git, Node.js), Basic HTML, CSS, and JavaScript for Frontend Development, Introduction to Backend Development with Node.js and Express			
Unit II	Developing the Backend	(8 Hours)	

Building RESTful APIs with Node.js and Express, Database Integration (Using a NoSQL Database like MongoDB), Implementing CRUD Operations, User Authentication and Authorization

Unit III	Developing the Frontend	(8 Hours)
-----------------	--------------------------------	------------------

Building Interactive Web Interfaces with React (or another modern framework/library), State Management in Single Page Applications (SPAs), Connecting the Frontend with Backend APIs, Implementing Responsive Web Design

Unit IV	Full Stack Integration and Deployment	(8 Hours)
----------------	--	------------------

Integrating Frontend and Backend Components, Web Application Security Best Practices, Performance Optimization and Testing, Deployment to a Cloud Platform (e.g., Heroku, AWS, or Firebase), Project: Designing and Developing a Full Stack Web Application

Learning Resources

Text Books:

Full-Stack React Projects: Learn MERN stack development by building modern web apps using MongoDB, Express, React, and Node.js; ShamaHoque; Packt Publishing Limited; 2nd edition

Reference Books:

1. Beginning MERN Stack: Build and Deploy a Full Stack MongoDB, Express, React, Node.js App; Greg Lim
2. Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node; Vasanth Subramanian; Apress; 2nd edition

Course Articulation Matrix (CO-PO Mapping) [Full Stack Web Development Laboratory -23CSE2117]															
COs	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	Relevant to National/Global/Regional
CO-1	3	--	--	--	3	--	--	--	--	--	--	--	2	2	National/Global
CO-2	3	2	2		3	--	--	--	--	--	--	--	2	2	National/Global
CO-3	--	2	2	2	3	--	--	--	--	--	--	--	2	2	National/Global
CO-4	2	2	--	2	3	--	--	--	--	--	--	--	2	2	National/Global
CO-5	--	2	2	2	3	--	--	--	--	--	--	--	2	2	National/Global
3 – HIGH, 2 – MEDIUM, 1 – LOW															

SEMESTER III

Course Name			
Economics, Politics and Rural Society Development			
Course Code:	23CSE3111	Course Credits:	3
Teaching Hours / Week (L: T: P):	2:1:0	CA Marks:	50
Total Number of Teaching Hours:	48	END-SEM Marks:	50
Course Objectives: (3 to 5)			
The main objectives of the course are:			
<ol style="list-style-type: none"> 1. Economic Analysis: Apply economic principles to analyze and interpret real-world economic phenomena. 2. Political Technology: Examine the influence of technology on contemporary political systems and processes. 3. Critical Problem-solving: Utilize critical thinking skills to address complex challenges in economics, politics, and rural development. 4. Stakeholder Communication: Communicate effectively with diverse stakeholders to facilitate informed decision-making in rural development contexts. 5. To build the healthy relationships with stake holder. 			
Course Outcomes:			
On completion of the course, students will be able:			
<ol style="list-style-type: none"> 1. To demonstrate an understanding of basic economic principles and their applications to real-world scenarios. 2. To analyze political systems and the role of technology in modern political processes. 3. To evaluate the challenges and opportunities in rural development, with a focus on the role of technology. 4. To apply economic and political principles to rural development scenarios. 5. To develop critical thinking skills to identify and solve complex problems related to economics, politics, and rural development. 			
Unit I	Introduction to Economics	(12 hours)	
Basic concepts of economics, including demand, supply, and market equilibrium; Macroeconomic concepts, such as GDP, inflation, and unemployment; Economic policies, such as fiscal and monetary policies; Applications of economics in the technology industry.			
Unit II	Political Systems and Processes	(12 Hours)	
Understanding of the functioning of political systems, such as democracy, authoritarianism, and socialism; The role of technology in modern political processes, including social media and big data analytics; Political policies and their impact on the economy and society.			

Unit III	Rural Development	(12 Hours)
Definition and characteristics of rural areas; Challenges and opportunities in rural development, including poverty, health, and education; Role of technology in rural development, including e-governance, e-commerce, and digital literacy; Policies and initiatives for rural development.		
Unit IV	Interdisciplinary Perspectives and Case Studies	(12 Hours)
Interdisciplinary approach to solving complex problems related to economics, politics, and rural development; Case studies of successful rural development initiatives that integrate technology, economics, and politics.		
Learning Resources		
<p>Text Books:</p> <ol style="list-style-type: none"> 1. Economics: Principles, Problems, and Policies by Campbell McConnell, Stanley Brue, and Sean Flynn. (Latest edition, published by McGraw-Hill Education India) 2. Politics: An Introduction by Andrew Heywood. (Latest edition, published by Palgrave Macmillan India) 3. Rural Development: Principles, Policies and Management by B. P. Vani and K. Sivaramane. (Latest edition, published by SAGE Publications India Pvt Ltd) <p>Reference Books:</p> <ol style="list-style-type: none"> 1. Rural Development: Putting the last first; Robert Chambers; Pearson Education India; 2nd Edition (2010) 2. Economic Development of Rural Areas: An Analysis of Problems and Prospects; L. P. Singh; Sage Publications India Pvt Ltd; 2nd Edition (2011) 		

Course Articulation Matrix (CO-POMapping) [Design for Developers -23CSE1117]															
COs	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	Relevant to National/Global/Regional
CO-1		2	2	2		--	--	--	--	--	--	--		2	National/Global
CO-2	2	2	3	3	2	--	--	--	--	--	--	--		2	National/Global
CO-3	2		2	3	3	--	--	--	--	--	--	--		2	National/Global
CO-4	--	2	2		2	--	--	--	--	--	--	2		2	National/Global
CO-5	--	--	--	--	3	2	--	--	--	--	--	--		2	National/Global
3 – HIGH, 2 – MEDIUM, 1 – LOW															

Course Name Mathematical Thinking 101			
Course Code:	23CSE3112	Course Credits:	4
Teaching Hours / Week (L: T: P):	3:1:0	CA Marks:	50
Total Number of Teaching Hours:	64	END-SEM Marks:	50
Course Objectives: (3 to 5) The main objectives of the course are: <ol style="list-style-type: none"> 1. Problem-solving Aptitude: Apply mathematical concepts proficiently to tackle challenges encountered in aptitude tests and technology-related industries. 2. Data Evaluation: Utilize mathematical methods to evaluate data systematically, facilitating informed decision-making. 3. Statistical Analysis: Analyze and interpret statistical data using mathematical techniques. 4. Conceptual Integration: Identify and integrate relationships between diverse mathematical concepts to solve problems effectively. 5. Demonstration capability: Whatever we have think method we must have to demonstrate 			
Course Outcomes: On completion of the course, students will be able: <ol style="list-style-type: none"> 1. To apply mathematical concepts to solve problems encountered in aptitude tests and the tech industry 2. To use mathematical methods to evaluate data and make informed decisions 3. To analyze and interpret statistical data 4. To understand the basic principles of geometry 5. To demonstrate proficiency in mental math and calculation 			
Unit I	Speed Math and Number system	(16 Hours)	
Mental calculation techniques for addition, subtraction, multiplication, and division; Approximation techniques for finding the nearest answer quickly; Tricks for working with squares, cubes, and square roots, Conversion between number systems: decimal, binary, octal, and hexadecimal Basic arithmetic operations in different number systems, Applications of number systems in computer science			
Unit II	Ratios, mixtures and allegations, Profit & Loss	(16 Hours)	
Concepts of ratios and proportions; Mixtures and Alligations concepts, Understanding profit and loss in business context; Applications of profit, loss, partnerships; Mixtures and Alligations problem-solving			
Unit III	Time and Work, Time Speed and Distance	(16 Hours)	

Concepts of time and work; Work efficiency and work-days formulas; Applications of time and work in problem-solving, Concepts of time, speed, and distance; Different formulas and their applications; Relative speed concept

Unit IV	Percentages, SI, CI, Permutation and Statistics	(16 Hours)
----------------	--	--------------------

Concepts of permutation; Circular permutations and permutations with restrictions; Applications of permutation in problem-solving, Concepts of percentage, simple interest, and compound interest; Different formulas and their applications; Installments and recurring deposits, Basic concepts of statistics: mean, median, mode, variance, and standard deviation; Data presentation: pie chart, bar chart, and line graph; Applications of statistics in problem-solving

Learning Resources

Text Books:

Quantitative Aptitude for Competitive Examinations; R.S. Aggarwal; S. Chand Publishing; 2021

Reference Books:

1. How to Prepare for Quantitative Aptitude for CAT; Arun Sharma; McGraw Hill Education; 2022
2. The Pearson Guide to Quantitative Aptitude for Competitive Examinations; Dinesh Khattar; Pearson Education; 2021

Course Articulation Matrix (CO-PO Mapping) [Design for Developers -23CSE1117]

COs	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	Relevant to National/Global/Regional
CO-1	2	2	--	--	--	--	--	--	--	--	--	--	2	2	National/Global
CO-2	--	3	3	--	--	--	--	--	--	--	--	--	2	2	National/Global
CO-3	--	--	--	--	--	2	--	--	--	--	2	--	2	2	National/Global
CO-4	2	2	--	--	3	--	--	--	--	--	--	--	2	2	National/Global
CO-5	--	3	2	--	--	--	--	--	--	--	--	--	2	2	National/Global

3 – HIGH, 2 – MEDIUM, 1 – LOW

Course Name			
Database Management Systems			
Course Code:	23CSE3113	Course Credits:	4
Teaching Hours / Week (L: T: P):	4:0:0	CA Marks:	50
Total Number of Teaching Hours:	51	END-SEM Marks:	50
Course Objectives: (3 to 5)			
The main objectives of the course are:			
<ol style="list-style-type: none"> 1. Database Concepts Understanding: Develop a deep understanding of fundamental concepts in database management systems, including data models, normalization principles, and database design methodologies. 2. Relational Database Design Proficiency: Demonstrate proficiency in designing normalized relational databases using SQL, ensuring adherence to normalization principles and best practices. 3. Theoretical Analysis of DBMS Tools: Analyze the theoretical foundations of popular DBMS tools such as MySQL, examining their roles in database creation, management, and optimization. 4. Advanced SQL Querying Skills: Master advanced SQL querying techniques to retrieve, manipulate, and manage data stored in relational databases, with a focus on theoretical underpinnings. 5. Problem-Solving with Theoretical Knowledge: Apply theoretical knowledge of database management concepts to analyze and solve complex real-world problems, fostering critical thinking and problem-solving abilities. 			
Course Outcomes:			
On completion of the course, students will be able:			
<ol style="list-style-type: none"> 1. To develop a solid understanding of the fundamental concepts of database management systems, encompassing data models, normalization, and database design principles. 2. To analyze and evaluate theoretical aspects of relational databases and SQL, focusing on conceptual clarity and theoretical foundations. 3. To examine the theoretical underpinnings of popular DBMS tools, such as MySQL, and their role in database creation and management. 4. To critically assess SQL as a query language and its theoretical basis for data retrieval, manipulation, and management. 5. To apply theoretical knowledge to analyze and propose solutions to real-world scenarios, emphasizing problem-solving skills within a theoretical framework. 			
Unit I	Introduction to DBMS	(13 Hours)	
The Evolution of Database Systems- Overview of a Database Management System-Outline of Database System Studies-The Entity-Relationship Data Model: Elements of the E/R Model-Design Principles-The Modeling of Constraints-Weak Entity Sets.			
Unit II	The Relational data model and Algebra	(13 Hours)	
Basics of the Relational Model-From E/R Diagrams to Relational Designs-Converting Subclass Structures to Relations-Functional Dependencies-Rules About Functional Dependencies-Design of Relational Database Schemas – Multi valued Dependencies- Relational Algebra: Relational operations-Extended Operators of Relational Algebra- Constraints on Relations.			

Unit III	SQL	(13 Hours)
Simple Queries in SQL-Sub Queries-Full-Relation Operations-Database Modifications-Defining a Relation Schema-View Definitions- Constraints and Triggers: Keys and Foreign Keys-Constraints on Attributes and Tuples-Modification of Constraints-Schema-Level Constraints and Triggers -Java Database Connectivity Security and User Authorization in SQL		
Unit IV	Index structures and query processing	(13 Hours)
Index Structures: Indexes on Sequential Files-Secondary Indexes-B-Trees-Hash Tables-Bitmap Indexes-Query Execution: Physical-Query-Plan Operators-One-Pass , two-pass & index based Algorithms, Buffer Management, Parallel Algorithms-Estimating the Cost of Operations-Cost-Based Plan Selection -Order for Joins-Physical-Query-Plan		
Unit V	Failure recovery and concurrency control	(12 Hours)
Issues and Models for Resilient Operation -Undo/Redo Logging-Protecting against Media Failures Concurrency Control: Serial and Serializable Schedules-Conflict-Serializability-Enforcing Serializability by Locks-Locking Systems With Several Lock Modes-Concurrency Control by Timestamps, validation transaction management: Serializability and Recoverability-View Serializability-Resolving Deadlocks Distributed Databases: commit& lock.		
Learning Resources		
Text Books:		
Database Systems: Models, Languages, Design And Application Programming By Ramez Elmasri, Shamkant B. Navathe, Pearson 6th edition		

Course Articulation Matrix(CO-PO Mapping)[Design for Developers -23CSE1117]															
COs	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	Relevant to National/Global/Regional
CO-1	2	2	--	3	2	--	--	--	--	--	--	--	2	2	National/Global
CO-2	--	2	--	2	3	--	--	--	--	--	--	2	--	2	National/Global
CO-3	--	--	--	2	3	--	--	--	--	--	--	--	2	3	National/Global
CO-4	--	--	--	2	3	--	--	--	--	2	2	--	2	2	National/Global
CO-5	2	--	2	2	2	--	--	--	--	--	--	--	2	--	National/Global
3 – HIGH, 2 – MEDIUM, 1 – LOW															

Course Name			
Object Oriented Programming			
Course Code:	23CSE3114	Course Credits:	3

Teaching Hours / Week (L: T: P):	3:0:0	CA Marks:	50
Total Number of Teaching Hours:	48	END-SEM Marks:	50
Course Objectives: (3 to 5)			
The main objectives of the course are:			
<ol style="list-style-type: none"> 1. OOP Understanding: Develop a comprehensive understanding of Object-Oriented Programming (OOP) principles, including encapsulation, inheritance, and polymorphism. 2. Analyzing OOP concepts: Analyze and evaluate the application of OOP concepts in software development using languages such as C++ and Python. 3. Design patterns: Evaluate and apply OOP design patterns and software engineering practices in the development of large-scale software projects. 4. Communication: Communicate effectively in a collaborative team environment, discussing and critiquing OOP-based software design and implementation. 5. Ability to develop modules. 			
Course Outcomes:			
On completion of the course, students will be able:			
<ol style="list-style-type: none"> 1. To explain the fundamental concepts and principles of OOP, such as encapsulation, inheritance, and polymorphism. 2. To apply OOP concepts to analyze and design software applications in C++ and Python. 3. To evaluate the performance and efficiency of OOP-based programs. 4. To design and implement complex data structures and algorithms using OOP concepts. 5. To analyze and critique the use of OOP design patterns and software engineering practices in large-scale software projects. 6. To collaborate and communicate effectively in a team environment to develop and maintain OOP-based software projects. 			
Unit I	Introduction to Object-Oriented Programming	(12 Hours)	
Basics of OOP, Objects and Classes, Abstraction, Encapsulation, Inheritance, Polymorphism, Procedural Programming vs OOP			
Unit II	Advanced Concepts in OOP	(12 Hours)	
Templates, Overloading, Exception Handling, Operator Overloading, Inheritance and Polymorphism, Advanced Topics in Inheritance			
Unit III	Object-Oriented Design Principles	(12 Hours)	
SOLID Principles, Design Patterns, Design for Reuse, Design for Testability			
Unit IV	Software Engineering Practices	(12 Hours)	

Agile Development, Waterfall Model, Software Development Life Cycle (SDLC), Version Control, Code Reviews, Testing and Debugging

Learning Resources

Text Books:

1. Object-Oriented Programming with C++; E Balagurusamy; McGraw Hill; Eighth edition
2. Python Object-Oriented Programming; Steven F. Lott & Dusty Phillips; Packt Publishing Limited; 4th edition
3. Java The Complete Reference; Herbert Schildt; McGraw Hill; Eleventh edition

Reference Books:

1. Object Oriented Programming C++; Robert Lafore; Pearson Education India; 4th edition
2. OOPS with C++ and Java; Balagurusamy; McGraw Hill Education 2014 edition
3. Python 3 Object-Oriented Programming; Dusty Phillips; Packt 3rd edition

Course Articulation Matrix (CO-PO Mapping) [Design for Developers -23CSE1117]

COs	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	Relevant to National/Global/Regional
CO-1	3	--	2	2	--	--	--	--	--	--	--	--	2	2	National/Global
CO-2	2	--	2	2	2	3	--	--	--	--	--	--	2	2	National/Global
CO-3	--	3	2	--	--	--	--	--	--	--	--	2	2	2	National/Global
CO-4	2	2	2	--	-	2	--	--	--	--	--	--	2	2	National/Global
CO-5	--	3	2	--	--	--	--	--	--	--	--	2	2	2	National/Global

3 – HIGH, 2 – MEDIUM, 1 - LOW

Course Name

Object Oriented Programming - Lab

Course Code:	23CSE3115	Course Credits:	1
Teaching Hours / Week (L: T: P):	0:0:2	CA Marks:	100
Total Number of Teaching	32	END-SEM Marks:	NA

Hours:		
<p>Course Objectives: (3 to 5) The main objectives of the course are:</p> <ol style="list-style-type: none"> 1. Application: To apply OOP concepts practically to develop software applications in C++ and Python. 2. Synthesis: To design and implement complex data structures and algorithms using OOP principles. 3. Collaboration: To collaborate effectively in a team environment to develop and maintain OOP-based software projects. 4. Evaluation: To evaluate the performance and efficiency of OOP-based programs through practical experimentation and analysis. 5. Problem-solving: To troubleshoot and debug OOP-based software applications effectively. 		
<p>Course Outcomes: On completion of the course, students will be able:</p> <ol style="list-style-type: none"> 1. To apply OOP principles practically to develop software applications in C++ and Python, demonstrating proficiency in implementation. 2. To design and implement complex data structures and algorithms using OOP concepts, emphasizing practical problem-solving skills. 3. To collaborate effectively in a team environment to develop and maintain OOP-based software projects, showcasing teamwork and communication skills. 4. To evaluate the performance and efficiency of OOP-based programs through practical experimentation and analysis. 5. To troubleshoot and debug OOP-based software applications effectively, demonstrating problem-solving abilities. 		
Unit I	Introduction to Object-Oriented Programming	(10 Hours)
Overview of OOP principles: encapsulation, inheritance, and polymorphism, Introduction to programming languages supporting OOP, Basic syntax and concepts: classes, objects, methods, and attributes, Hands-on exercises: Implementing simple OOP concepts		
Unit II	Advanced OOP Concepts and Design Patterns	(10 Hours)
In-depth study of advanced OOP concepts: abstraction, encapsulation, inheritance, and polymorphism, Introduction to design patterns: creational, structural, and behavioral patterns, Practical application of design patterns in software development, Hands-on exercises: Implementing design patterns in projects		
Unit III	Project Development and Software Engineering Practices	(12 Hours)
Overview of project development lifecycle: planning, design, implementation, testing, and maintenance, Introduction to software engineering practices: version control, testing methodologies, documentation, and code reviews, Team-based project development: Collaborative development of OOP-based software projects, Final project presentation and evaluation: Demonstration of OOP principles, design patterns, and software engineering practices in a real-world project		
Learning Resources		

Text Books:

1. Object-Oriented Programming with C++; E Balagurusamy; McGraw Hill; Eighth edition
2. Python Object-Oriented Programming; Steven F. Lott & Dusty Phillips; Packt Publishing Limited; 4th edition
3. Java The Complete Reference; Herbert Schildt; McGraw Hill; Eleventh edition

Reference Books:

1. Object Oriented Programming C++; Robert Lafore; Pearson Education India; 4th edition
2. OOPS with C++ and Java; Balagurusamy; McGraw Hill Education 2014 edition
3. Python 3 Object-Oriented Programming; Dusty Phillips; Packt 3rd edition

Course Articulation Matrix (CO-PO Mapping) [Design for Developers -23CSE1117]															
COs	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	Relevant to National/Global/Regional
CO-1	2	3	--	--	--	--	--	--	--	--	2	2	2	2	National/Global
CO-2	2	2	2	--	--	--	--	2	---	--	--	--	2	2	National/Global
CO-3	2	2	--	--	--	---	2	2	--	--	--	--	2	2	National/Global
CO-4	--	3	3	--	--	--	---	--	--	--	--	--	2	2	National/Global
CO-5	--	3	3	--	--	--	--	--	--	--	--	--	2	2	National/Global
3 – HIGH, 2 – MEDIUM, 1 – LOW															

Course Name			
Database Management Systems – Lab			
Course Code:	23CSE3116	Course Credits:	1
Teaching Hours / Week (L: T: P):	0:0:2	CA Marks:	100
Total Number of Teaching Hours:	32	END-SEM Marks:	NA

Course Objectives: (3 to 5)

The main objectives of the course are:

1. **SQL Proficiency:** Develop proficiency in SQL by writing and executing queries to retrieve, manipulate, and manage data stored in relational databases.
2. **Database Design Skills:** Gain practical experience in designing and implementing normalized relational databases, adhering to normalization principles and best practices.
3. **Advanced SQL Knowledge:** Acquire advanced knowledge of SQL concepts such as joins, subqueries, views, triggers, and stored procedures, enabling complex data manipulation and management tasks.
4. **Transaction Management Competence:** Understand and apply transaction management concepts, including the maintenance of ACID properties, concurrency control, and locking mechanisms.
5. **Database Administration Abilities:** Learn essential database administration tasks such as backup and recovery, user management, and performance tuning to ensure the reliability, security, and efficiency of database systems.

Course Outcomes:

On completion of the course, students will be able:

1. To develop practical skills in database design and implementation using SQL, focusing on real-world applications and industry standards.
2. To design and implement normalized relational databases using SQL, ensuring data integrity, efficiency, and scalability.
3. To utilize popular DBMS tools like MySQL to create, manage, and optimize databases, gaining hands-on experience in database administration tasks.
4. To master SQL querying techniques to extract, manipulate, and manage data stored in relational databases efficiently, enhancing data retrieval and analysis capabilities.
5. To develop functional, database-driven web applications by applying database management concepts to practical scenarios, showcasing proficiency in database integration and application development.
6. To synthesize practical knowledge and theoretical understanding to develop innovative database solutions, demonstrating problem-solving skills and readiness for real-world database management challenges.

Unit I	Introduction to Database Concepts and SQL	(8 Hours)
Overview of Database Management Systems (DBMS) and their importance; Introduction to Relational Database Management Systems (RDBMS) and SQL; Basic SQL commands: SELECT, INSERT, UPDATE, DELETE; Hands-on exercises: Writing and executing SQL queries in a database environment		
Unit II	Database Design and Normalization	(8 Hours)
Principles of database design: entity-relationship modeling, normalization; Normal forms and normalization techniques (1NF, 2NF, 3NF); Database schema design and entity-relationship diagrams (ERDs); Hands-on exercises: Designing and normalizing databases using SQL		
Unit III	Advanced SQL and Transaction Management	(8 Hours)

Advanced SQL concepts: JOINS, subqueries, views, triggers, stored procedures; Transaction management: ACID properties, concurrency control, and locking; Practical applications of advanced SQL and transaction management; Hands-on exercises: Implementing advanced SQL queries and transaction management techniques

Unit IV	Database Administration and Project Development	(8 Hours)
Database administration tasks: backup and recovery, user management, performance tuning, Introduction to database security and access control, Team-based project development: Collaborative development of database-driven applications		

Learning Resources

Text Books:

Database Systems: Models, Languages, Design And Application Programming By Ramez Elmasri, Shamkant B. Navathe, Pearson 6th edition

Course Articulation Matrix (CO-PO Mapping) [Design for Developers -23CSE1117]															
COs	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	Relevant to National/Global/Regional
CO-1	--	--	--	3	2	--	--	--	--	--	--	--	2	2	National/Global
CO-2	--	2	2	--	--	--	--	--	--	--	--	--	2	2	National/Global
CO-3	--	--	2	3	--	--	--	--	2	3	--	--	2	2	National/Global
CO-4	2	2	3	--	--	--	--	--	--	--	--	--	2	2	National/Global
CO-5	2	2	2	--	--	2	--	--	--	--	--	--	2	2	National/Global
3 – HIGH, 2 – MEDIUM, 1 – LOW															

SEMESTER IV

Course Name			
How human languages work			
Course Code:	23CSE4111	Course Credits:	4
Teaching Hours / Week (L: T: P):	3:1:0	CA Marks:	50
Total Number of Teaching Hours:	64	END-SEM Marks:	50
Course Objectives: (3 to 5)			
The main objectives of the course are:			
<ol style="list-style-type: none"> 1. Linguistic Foundations: Develop a foundational understanding of human language by comprehending basic concepts in phonetics, syntax, semantics, and pragmatics. 2. Language Structure Exploration: Explore the structure and function of human language, delving into the sound systems, grammatical structures, and meanings of words and sentences across languages. 3. Social and Cultural Insights: Gain insight into the social and cultural dimensions of language use, examining language variation, multilingualism, and language attitudes to understand their impact on society. 4. Neurocognitive Perspective: Understand the neural mechanisms of language processing in the brain, exploring how language comprehension and production are facilitated and processed. 			
Course Outcomes:			
On completion of the course, students will be able:			
<ol style="list-style-type: none"> 1. To understand the basic concepts of human language, including phonetics, syntax, semantics, and pragmatics. 2. To explore the structure and function of human language, including the sound systems of languages, grammatical structures, and the meanings of words and sentences. 3. To understand the social and cultural dimensions of language use, including language variation and change, multilingualism, and language attitudes. 4. To understand how language is processed in the brain, including the neural mechanisms of language comprehension and production. 5. To analyze the ways in which language reflects and shapes cultural and social identities. 6. To develop critical thinking skills by analyzing linguistic data and applying linguistic concepts to real-world problems. 			
Unit I	Introduction to Language and Linguistics	(16 hours)	
Overview of Linguistics and its sub-fields, Phonetics and Phonology, Morphology			
Unit II	Syntax and Semantics	(16 Hours)	
Syntax and Syntactic Structures, Semantics and Semantic Structures, Language Typology			
Unit III	Language Acquisition and Language Change	(16 Hours)	
First and Second Language Acquisition, Language Change and Language Contact, Historical Linguistics			

Unit IV	Sociolinguistics and Applied Linguistics	(16 Hours)
Sociolinguistics and Language Variation, Language Policy and Planning, Language and Identity, Language and Technology, Language and Culture, Language and Gender, Language and Power, Language and Globalization		
Learning Resources		
Text Books: How Languages Work: An Introduction to Language and Linguistics by Carol Genetti; Cambridge University Press; 2nd edition		
Reference Books: How Language Works; David Crystal; Penguin UK (2007 edition)		

Course Articulation Matrix (CO-PO Mapping) [Design for Developers -23CSE1117]															
COs	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	Relevant to National/Global/Regional
CO-1	2	2	2	--	--	--	--	--	--	--	--	2	2	2	National/Global
CO-2	2	2	2	--	--	--	--	--	--	3	3	--	2	2	National/Global
CO-3	--	3	3	--	--	--	--	--	--	--	2	--	2	2	National/Global
CO-4	--	--	--	--	2	2	--	--	--	--	--	--	2	2	National/Global
CO-5	2	3	--	--	3	3	--	--	--	--	--	--	2	2	National/Global
3 – HIGH, 2 – MEDIUM, 1 – LOW															

<p>Course Name Tools and techniques for Creative thinking</p>
--

Course Code:	23CSE4112	Course Credits:	4
Teaching Hours / Week (L: T: P):	3:1:0	CA Marks:	50
Total Number of Teaching Hours:	64	END-SEM Marks:	50
<p>Course Objectives: (3 to 5)</p> <p>The main objectives of the course are:</p> <ol style="list-style-type: none"> 1. Creative Idea Generation: Utilize a variety of creative thinking tools and techniques to generate innovative ideas across different domains. 2. Critical Analysis and Evaluation: Apply critical thinking skills to analyze and evaluate creative ideas and solutions, discerning their feasibility and effectiveness. 3. Barriers Identification and Overcoming: Identify and overcome barriers to creative thinking and problem-solving, fostering a mindset conducive to innovation. 4. Practical Application of Creativity: Apply creative thinking skills in practical contexts, such as business, design, and technology, to address real-world challenges and opportunities. 5. work collaboratively with others to generate and implement creative solutions 			
<p>Course Outcomes:</p> <p>On completion of the course, students will be able:</p> <ol style="list-style-type: none"> 6. To use a variety of creative thinking tools and techniques to generate new ideas 7. To apply critical thinking skills to analyze and evaluate creative ideas and solutions 8. To identify and overcome barriers to creative thinking and problem-solving 9. To apply creative thinking skills in practical situations, such as in business, design, and technology 10. To develop effective communication skills for presenting and selling creative ideas 			
Unit I	Introduction to Principles of Creativity	(16 hours)	
Mother and father of innovation, Levels of creativity, Creative environments, Creativity tools, Brainstorming techniques, Principles of brainstorming, Flip chart, Post-it, Alphabet brainstorming, Brainwriting, Grid brainstorming			
Unit II	Thinking styles	(16 Hours)	
The value of diversity, Principles of various thinking styles, Design thinking, Different thinking styles in practice			
Unit III	Morphological analysis and TRIZ theory	(16 Hours)	
Principles of morphological analysis, Group application of plotline MA, Principles and discussion, Contradiction matrix, TRIZ Parameters and Principles			
Unit IV	SCAMPER and tools in combination	(16 Hours)	

SCAMPER for architecture, team innovation using SCAMPER, Use of different thinking styles, Creative problem solving, Double diamond model, Circle brainstorming steps, E-tivity: B-link

Learning Resources

Text Books:

Lateral Thinking: A Textbook of Creativity; Penguin; 2016 edition

Reference Books:

Cracking Creativity: The Secrets of Creative Genius" by Michael Michalko (2001); Ten Speed Press; Revised edition (26 June 2001)

"A Whack on the Side of the Head: How You Can Be More Creative" by Roger von Oech (2008); Grand Central Publishing; Special edition (5 May 2008)

Course Articulation Matrix (CO-PO Mapping) [Design for Developers -23CSE1117]

COs	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	Relevant to National/Global/Regional
CO-1	--	2	3	--	--	--	--	--	--	--	--	3	2	2	National/Global
CO-2	2	--	2	3	3	--	--	--	--	--	--	--	2	2	National/Global
CO-3	--	2	2	2	3	--	--	--	--	--	--	--	2	2	National/Global
CO-4	2	2	2	--	3	--	--	--	--	--	--	3	2	2	National/Global
CO-5	--	2	--	2	2	--	--	--	--	--	--	--	2	2	National/Global

3 – HIGH, 2 – MEDIUM, 1 - LOW

Course Name
Operating Systems

Course Code:

23CSE4113

Course Credits:

3

Teaching Hours / Week (L: T: P):	3:0:0	CA Marks:	50
Total Number of Teaching Hours:	48	END-SEM Marks:	50
<p>Course Objectives: (3 to 5)s The main objectives of the course are:</p> <ol style="list-style-type: none"> 1. Fundamental Concepts Understanding: Develop a comprehensive understanding of fundamental operating system concepts, including processes, threads, synchronization, and memory management. 2. Algorithm Performance Analysis: Analyze the performance of various scheduling and memory allocation algorithms, evaluating their efficiency and effectiveness in different scenarios. 3. Operating System Comparison: Compare and contrast different types of operating systems, such as batch, multi-programmed, and real-time systems, to understand their features, advantages, and limitations. 4. Practical Implementation and Security Awareness: Gain practical experience in implementing basic operating system functionalities while also developing an awareness of security and privacy issues inherent in modern operating systems. 			
<p>Course Outcomes: On completion of the course, students will be able:</p> <ol style="list-style-type: none"> 1. To understand the fundamental concepts of operating systems such as processes, threads, synchronization, and memory management. 2. To analyze the performance of various scheduling and memory allocation algorithms. 3. To compare and contrast different types of operating systems, such as batch, multi-programmed, and real-time systems. 4. To develop an understanding of device management, file systems, and virtualization. 5. To gain practical experience in implementing basic operating system functionalities. 6. To understand the security and privacy issues in modern operating systems. 			
Unit I	Computer Arithmetic & Processor Organisation	(16 hours)	
Computer Registers, Classification of Instruction – Size: three, two, one and zero instruction, Addressing Mode. Arithmetic and Logic Circuit Design. Instruction execution cycle: Sequencing of control signals, hardwired control, micro-programmed control, control signals, microinstructions, micro program sequencing, pre-fetching microinstructions. Introduction to graphical processing unit (GPU).			
Unit II	Memory Organization	(16 Hours)	
Memory hierarchy, Main memories chip architectures, memory address map, memory assembly to CPU. Auxiliary memory: magnetic tapes, disks (magnetic and SSDs). Associate memory: hardware organization, match logic, read and writes operations. Cache memory. Memory interleaving technique.			
Unit III	Parallel Processing	(16 Hours)	

Parallel processing, examples of parallel processing machines. Classification of parallel processing: Handler classification – pipeline processing, vector processing and array processing, Flynn’s classification – SISD, SIMD, MISD, MIMD. Pipeline conflicts.

Unit IV	Process Management	(16 Hours)
----------------	---------------------------	--------------------

Introduction to operating systems. Process Management: what is a Process?, Process state, Process control block. Threads. Cooperating processes. Inter-process communication. CPU scheduling algorithms: First come first serve, shortest job first – primitive & non primitive, Round Robin. Deadlock: Necessary conditions for occurrence of deadlocks, Deadlock detection – Resource Allocation Graph. Deadlock Avoidance Algorithms: Banker Algorithm and Safety Algorithm.

Learning Resources

Text Books:

Operating Systems Concepts, Abraham Silberschatz, Peter B. Galvin and Greg Gagne, Wiley, 2012.

Reference Books:

1. The Design of the Unix Operating System, Maurice Bach, Pearson; 1st edition
2. Operating systems concepts; Avi Silberschatz, Peter Baer Galvin, Greg Gagne; Wiley; Ninth edition

Course Articulation Matrix (CO-PO Mapping) [Design for Developers -23CSE1117]															
COs	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	Relevant to National/Global/Regional
CO-1	--	2	2	3	--	--	2	--	--	---	---	--	2	2	National/Global
CO-2	--	--	3	3	--	--	--	--	--	---	---	--	2	2	National/Global
CO-3	2	2	2	--	2	--	--	--	--	---	---	--	2	2	National/Global
CO-4	-	2	2	2	3	-	--	-	-	-	-	--	2	2	National/Global
CO-5	2	2	2		3	-	-	-	-	-	-	--	2	2	National/Global
3 – HIGH, 2 – MEDIUM, 1 - LOW															

Course Name			
Data Structures and Algorithms			
Course Code:	23CSE4114	Course Credits:	3
Teaching Hours / Week (L: T: P):	3:0:0	CA Marks:	50
Total Number of Teaching Hours:	48	END-SEM Marks:	50

Course Objectives: (3 to 5)

The main objectives of the course are:

1. **Conceptual Understanding:** Develop a comprehensive understanding of the principles and concepts underlying data structures and algorithms, including complexity analysis, recursion, and graph theory.
2. **Design and Implementation Proficiency:** Design and implement efficient data structures and algorithms using various techniques and approaches, focusing on optimization and effectiveness.
3. **Performance Analysis Skills:** Analyze the performance and correctness of data structures and algorithms using both mathematical analysis and empirical methods, enabling informed decision-making.
4. **Real-World Application:** Apply data structures and algorithms to solve real-world problems across various domains such as search, sorting, and graph algorithms, demonstrating practical problem-solving skills.

Course Outcomes:

On completion of the course, students will be able:

1. To explain the principles and concepts underlying data structures and algorithms, such as complexity analysis, recursion, and graph theory.
2. To design and implement efficient data structures and algorithms, using a variety of techniques and approaches.
3. To analyze the performance and correctness of data structures and algorithms, using mathematical and empirical methods.
4. To apply data structures and algorithms to solve real-world problems, across a range of application areas such as search, sorting, and graph algorithms.
5. To evaluate different data structures and algorithms, based on their suitability for specific problem domains and constraints.
6. To create and present well-structured and well-documented code that implements data structures and algorithms.

Unit I	Introduction	(12 hours)
Algorithmic thinking, peak finding, Models of computation, Document distance, Python cost model		
Unit II	Sorting and trees	(12 Hours)
Insertion sort, Merge sort, Heaps and heap sort, Binary search trees, BST sort, AVL trees and sort, Counting sort, radix sort, lower bounds for sorting and searching		
Unit III	Hashing	(12 Hours)
Hashing with chaining, Table doubling, Karp-Rabin, Open addressing, Cryptographic hashing		
Unit IV	Numerics and Graphs	(12 Hours)

Integer arithmetic, Karatsuba multiplication, Square roots, Newton's method, Integer arithmetic, Karatsuba multiplication, Square roots, Newton's method

Learning Resources

Text Books:

Data Structures and Algorithms Made Easy; NarasimhaKarumanchi; CareerMonk Publications; 5th edition

Reference Books:

Introduction to Algorithms; Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein; PHI Learning Pvt. Ltd. (Originally MIT Press); Third edition

Course Articulation Matrix (CO-PO Mapping) [Design for Developers -23CSE1117]

COs	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	Relevant to National/Global/Regional
CO-1	-	2	2	3	-	-	-	-	-	-	-	-	2	2	National/Global
CO-2	2	3	3		2	-	-	-	-	-	-	-	2	2	National/Global
CO-3	2	-	-	2	2	3	-	-	-	-	-	-	2	2	National/Global
CO-4	-	2	2	3	-	-	-	-	-	-	-	-	2	2	National/Global
CO-5	-	2	2	3	3	-	-	-	-	-	-	-	2	2	National/Global

3 – HIGH, 2 – MEDIUM, 1 - LOW

Course Name

Operating Systems - Lab

Course Code:	23CSE4115	Course Credits:	1
Teaching Hours / Week (L: T: P):	0:0:2	CA Marks:	100
Total Number of Teaching Hours:	32	END-SEM Marks:	NA

Course Objectives: (3 to 5)

The main objectives of the course are:

1. Mastering Basic Commands: Execute fundamental operating system commands proficiently, demonstrating mastery of basic system operations and utilities.
2. Practical Application of Concepts: Apply theoretical knowledge of operating system concepts to diagnose and resolve practical issues encountered in lab exercises, fostering problem-solving skills.
3. Effective System Administration: Implement system administration tasks effectively, such as configuring network settings and managing user accounts, showcasing practical skills in system management.
4. Automation through Scripting: Develop and deploy shell scripts to automate routine tasks, showcasing creativity and efficiency in task automation and script development.

Course Outcomes:

On completion of the course, students will be able to:

1. Understand and execute fundamental operating system commands, demonstrating procedural knowledge.
2. Apply operating system concepts to practical scenarios, analyzing and troubleshooting common issues.
3. Implement system administration tasks including backup, recovery, and network configurations.
4. Develop shell scripts to automate routine tasks, demonstrating creativity and problem-solving skills.
5. Collaborate effectively with peers to solve complex operating system problems, fostering teamwork and communication.
6. Present lab activities and findings clearly, and document procedures effectively, synthesizing knowledge and demonstrating communication skills.

Unit I	Computer Arithmetic & Processor Organization	(8 hours)
Computer Registers, Classification of Instruction Sizes: three, two, one, and zero instruction, Addressing Modes, Arithmetic and Logic Circuit Design, Instruction Execution Cycle, Control Signals and Microinstructions, Microprogrammed Control and Control Signal Sequencing, Introduction to Graphical Processing Unit (GPU)		
Unit II	Memory Organization	(8 Hours)
Memory Hierarchy, Main Memory Chip Architectures, Memory Address Map, Memory Assembly to CPU, Auxiliary Memory: Magnetic Tapes, Disks (Magnetic and SSDs), Associate Memory: Hardware Organization, Match Logic, Read and Write Operations, Cache Memory, Memory Interleaving Technique		
Unit III	Parallel Processing	(8 Hours)
Parallel Processing, Examples of Parallel Processing Machines, Classification of Parallel Processing: Handler Classification - Pipeline Processing, Vector Processing, and Array Processing, Flynn's Classification - SISD, SIMD, MISD, MIMD, Pipeline Conflicts		
Unit IV	Introduction to Operating Systems and Process	(8 Hours)

Management
Introduction to Operating Systems, Process Management: Process Basics, Process State, Process Control Block, Threads, Cooperating Processes, Inter-process Communication, CPU Scheduling Algorithms: First Come First Serve, Shortest Job First (Primitive & Non-Primitive), Round Robin, Deadlock: Necessary Conditions for Occurrence, Deadlock Detection - Resource Allocation Graph, Deadlock Avoidance Algorithms: Banker Algorithm and Safety Algorithm
Learning Resources
Text Books: Operating Systems Concepts, Abraham Silberschatz, Peter B. Galvin and Greg Gagne, Wiley, 2012.
Reference Books: The Design of the Unix Operating System, Maurice Bach, Pearson; 1st edition Operating systems concepts; Avi Silberschatz, Peter Baer Galvin, Greg Gagne; Wiley; Ninth edition

Course Articulation Matrix (CO-PO Mapping) [Design for Developers -23CSE1117]															
COs	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	Relevant to National/Global/Regional
CO-1	-	2	2	2	3	-	-	-	-	-	-	-	2	2	National/Global
CO-2	-	3	3	2	2	-	-	-	-	-	-	-	2	2	National/Global
CO-3	2	2	3	-	-	-	-	-	-	-	-	-	2	2	National/Global
CO-4	-	2	2	2	3	-	-	-	-	-	-	-	2	2	National/Global
CO-5	3	2	1	4	5	-	-	-	-	-	-	-	2	2	National/Global
3 – HIGH, 2 – MEDIUM, 1 – LOW															

Course Name			
Data Structures and Algorithms – Lab			
Course Code:	23CSE4116	Course Credits:	1
Teaching Hours / Week (L: T: P):	0:0:2	CA Marks:	100
Total Number of Teaching Hours:	32	END-SEM Marks:	NA

Course Objectives: (3 to 5)

The main objectives of the course are:

1. Apply Theoretical Concepts Practically: Implement data structures and algorithms learned in theory classes into practical coding exercises to deepen understanding and proficiency.
2. Design and Implement Efficient Solutions: Develop the ability to design and implement efficient solutions for various computational problems using appropriate data structures and algorithmic techniques.
3. Evaluate and Analyze Performance: Analyze the performance and correctness of implemented solutions using mathematical analysis and empirical testing, refining problem-solving skills.
4. Develop Proficient Coding Skills: Enhance coding skills by creating well-structured, documented, and optimized code solutions, fostering professionalism and proficiency in coding practices.

Course Outcomes:

On completion of the course, students will be able:

1. To apply theoretical concepts of data structures and algorithms to practical programming tasks.
2. To design and implement efficient solutions for various computational problems using different algorithmic techniques.
3. To assess the performance and correctness of implemented data structures and algorithms through experimentation and analysis.
4. To solve real-world problems using data structures and algorithms across diverse application domains.
5. To evaluate and compare the effectiveness of different data structures and algorithms in solving specific problem instances.
6. To produce well-structured and well-documented code solutions that adhere to best coding practices and standards.

Unit I	Introduction	(8 hours)
Problems based on - Algorithmic thinking, peak finding, Models of computation, Document distance, Python cost model		
Unit II	Sorting and trees	(8 Hours)
Problems based on - Insertion sort, Merge sort, Heaps and heap sort, Binary search trees, BST sort, AVL trees and sort, Counting sort, radix sort, lower bounds for sorting and searching		
Unit III	Hashing	(8 Hours)
Problems based on - Hashing with chaining, Table doubling, Karp-Rabin, Open addressing, Cryptographic hashing		
Unit IV	Numerics and Graphs	(8 Hours)

Problems based on - Integer arithmetic, Karatsuba multiplication, Square roots, Newton's method, Integer arithmetic, Karatsuba multiplication, Square roots, Newton's method

Learning Resources

Text Books:

Data Structures and Algorithms Made Easy; NarasimhaKarumanchi; CareerMonk Publications; 5th edition

Reference Books:

Introduction to Algorithms; Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein; PHI Learning Pvt. Ltd. (Originally MIT Press); Third edition

Course Articulation Matrix (CO-PO Mapping) [Design for Developers -23CSE1117]															
COs	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO-1	PSO-2	Relevant to National/Global/Regional
CO-1	-	2	2	-	3	-	-	-	-	-	-	-	2	2	National/Global
CO-2	3	3	2	-	-	-	-	-	-	-	-	-	2	2	National/Global
CO-3	2	2	2	3	-	-	-	-	-	-	-	-	2	2	National/Global
CO-4	--	--	2	2	-	-	-	-	-	-	-	-	2	2	National/Global
CO-5	-	2	2	2	3	-	-	----	-	-	-	-	2	2	National/Global
3 – HIGH, 2 – MEDIUM, 1 – LOW															