AC –
Item No. –

As Per NEP 2020

University of Mumbai

Syllabus for Major
Vertical – 1, 4 & 6

Name of the Programme – B.E. (Information Technology)		
Faulty of Engineering		
Board of Studies in Information Technology		
U.G. Second Year Programme	Exit Degree	U.G. Diploma in Engineering-Information Technology.
Semester		III & IV
From the Academic Year		2025-26

University of Mumbai

(As per NEP 2020)

Sr. No.	Heading	Particulars
1	Title of program O:____________________	B.E. (Information Technology)
2	Exit Degree	U.G. Diploma in Engineering-Information Technology.
3	Scheme of Examination R: ____________	NEP 40% Internal 60% External, Semester End Examination Individual Passing in Internal and External Examination
4	Standards of Passing R: ______________	40%
5	Credit Structure R: ______________	Attached herewith
6	Semesters	Sem. III & IV
7	Program Academic Level	5.00
8	Pattern	Semester
9	Status	New
10	To be implemented from Academic Year	2025-26

Sd/-
Dr. Vaishali D. Khairnar
BoS-Chairman-Information Technology
Faculty of Technology

Sd/-
Dr. Deven Shah
Associate Dean
Faculty of Science & Technology

Sd/-
Prof. Shivram S. Garje
Dean
Faculty of Science & Technology

Preamble

To meet the challenge of ensuring excellence and NEP 2020 policy in engineering education, the issue of quality needs to be addressed, debated, and taken forward systematically. Accreditation is the principal means of quality assurance in higher education. The major emphasis of the accreditation process is to measure the outcomes of the program that is being accredited. In line with this Faculty of Science and Technology (in particular Engineering) of the University of Mumbai has taken the lead in incorporating the philosophy of NEP 2020 education in the process of curriculum development.

The second-year engineering course is a core training program to impart scientific and logical thinking training to learners in general, with a choice of course selection from the program core course, multidisciplinary minor, and vocational skill-enhanced course. Simultaneously, the objectives of NEP 2020 demand nurturing the core program and skills required for the Information Technology Branch of engineering in the learner. Keeping this in view, a pool of courses is offered in Core Courses covering fundamentals required to understand core and modern engineering practices and emerging trends in technology. Considering the shift in pedagogy and the convenience of a stress-free learning process, a choice-based subject pool is offered in the coursework under the heads of Information Technology in Engineering for open electives and multidisciplinary minor courses in the third and fourth semesters. Essentially, to give a glimpse of trends in the industry under vocational and enhanced skill practices, the pool is offered to nurture and develop creative skills in contemporary industrial practices. Criteria met in the structure is the opportunity for learners to choose the course of their interest in all disciplines.

Program Core Course Cover Information Technology core courses. Also, OE and MDM where a pool of subjects are given for selection. Considering the present scenario, diverse choices need to be made available to fulfill the expectation of a learner to aspire for a career in the field of current trends of Technology and interdisciplinary research. Ability enhancement can be achieved in Undergraduate training by giving an objective viewpoint to the learning process and transitioning a learner from a rote learner to a creative professional. for the purpose Design Thinking is introduced in the First Semester to orient a journey learner to become a skilled professional. Considering the NEP-2020 structure of award of Certificate & Diploma at multiple exit-point pools of Vocational skills is arranged for giving exposure to the current Industry practices.

The faculty resolved that course objectives and course outcomes are to be clearly defined for every course so that all faculty members in affiliated higher education institutes understand the depth and approach of the course to be taught, which will enhance the learner's learning process. NEP 2020 grading system enables a much-required shift in focus from teacher-centric to continuous-based learner-centric education since the workload estimated is based on the investment of time in learning and not in teaching. It also focuses on continuous evaluation, which will enhance the quality of education. Credit assignment for courses is based on a 15-week teaching-learning process for NEP 2020, however, the content of courses is to be taught in 12-13 weeks, and the remaining 2-3 weeks are to be utilized for revision, tutorial, guest lectures, coverage of content beyond the syllabus, etc.

There was a concern that in the present system, the second-year syllabus must not be heavily loaded to the learner and it is of utmost importance that the learner entering into the second year of an engineering course should feel at ease by lowering the burden of syllabus and credits. This is necessary for a learner to get accustomed to the new environment of a college and to create a bond between the teacher and the learner. The present curriculum will be implemented for the Second Year of Engineering from the academic year 2025-26. Subsequently, this system will be carried forward for Third Year and Final Year Engineering in the academic years 2026-27, and 2027-28, respectively.

Sd/-
Dr. Vaishali D. Khairnar
BoS-Chairman-Information Technology
Faculty of Technology

Sd/-
Dr. Deven Shah
Associate Dean
Faculty of Science & Technology

Sd/-
Prof. Shivram S. Garje
Dean
Faculty of Science & Technology

Under Graduate Diploma in Engineering- Information Technology.

Credit Structure (Sem. III & IV)

	R: _________________C									
Level	Semester	Major		Minor	OE	VSC, SEC (VSEC)	AEC, VEC, IKS	OJT, FP, CEP, CC,RP	Cum. Cr. / Sem.	Degree/ Cum. Cr.
		Mandatory	Electives							
5.0	III	PCC301:3 PCC302:3 PCC303:3 PCC304:3 PCL301: 1 PCL302:1	--	--	OE:2	--	VEC: 2 HSL: 2	CEP: 2	22	UG Diploma 45
	R:\__________________D									
	IV	PCC401:3 PCC402:3 PCC403:3 PCL401:1 PCL402:1	--	MDM: 4	OE:2	VSEC:2	VEC: 2 EEM:2	--	23	
	Cum Cr.	25	--	4	4	2	*2+2+2+2*	2	45	

Exit option: Award of UG Diploma in Major and MDM with 90 credits and additional 4 credits core **one** theory subject with 3 credits and **one** lab with 1 credit from one third year from where they want to take Exit degree. Along with theory and practical course student must compulsory do internship for **one month or 160 hours** which internship is equal to 4 credits.

[Abbreviation – PCC- Program Core Course, PCL- Program Core Lab, PEC- Program Elective Course, PEL- Program Elective Lab, OE – Open Electives, VSEC- Vocational Skill Enhancement Course, [VSC – Vocation Skill Course, SEC – Skill Enhancement Course], HSSM- Humanities Social Science and Management, [AEC – Ability Enhancement Course, VEC – Value Education Course, IKS – Indian Knowledge System], Experiential Learning Courses [OJT – on Job Training, FP – Field Project, CEP – Comm. Engg. Project, CC – Co-Curricular, RP – Research Project]]

Sem. – III & IV

S.E.IT
Scheme

Program Structure for Second Year of Information Technology

UNIVERSITY OF MUMBAI (With Effect from 2025-2026)

SEMESTER III

Course Code	Course Description	Teaching Scheme (Contact Hours)			Credit Assigned			
		Theory	Practical	Tutorial	Theory	Tutorial	Practical	Total Credits
2343111	Applied Mathematics Thinking-I	2	--	1-	2	1	--	3
2343112	Advance Data Structure & Analysis	3	–	--	3	--	--	3
2343113	Database Management System & Application	3	--	--	3	--	--	3
2343114	Automata Theory	3	--	--	3	--	--	3
OEC301	Open Elective	2#	--	--	2	--	--	2
2343115	ADSA Lab	--	2	--	--	--	1	1
2343116	SQL Lab	--	2	--	--	--	1	1
2343611	Mini-Project - Full Stack Java Programming	–	2*+2	–	–	–	2	2
2993511	Entrepreneurship Development	--	2*+2	---	--	--	2	2
2993512	Environmental Science	--	2*+2	--	--	--	2	2
Total		**13**	**16**	**01**	**13**	**01**	**08**	**22**

* Two hours of practical class to be conducted for full class as demo/discussion.

Theory / Tutorial 1 credit for 1 hour and Practical 1 credit for 2 hours

# Institute shall offer a course for Open Elective from Science/Commerce/Management stream bucket provided by the University of Mumbai.

#Institute shall offer a course for MDM from other Engineering Boards.

Course Code	Course Description	Examination scheme							
		Internal Assessment Test (IAT)			End Sem. Exam Marks	End Sem. Exam Duration (Hrs)	Term Work (Tw)	Oral & Pract.	Total
		IAT-I	IAT-II	Total (IAT-I) + IAT-II)					
2343111	Applied Mathematics Thinking-I	20	20	40	60	2	25	--	125
2343112	Advance Data Structure & Analysis	20	20	40	60	2	--	--	100
2343113	Database Management System & Application	20	20	40	60	2	--	--	100
2343114	Automata Theory	20	20	40	60	2	--	--	100
OEC301	Open Elective	20	20	40	60	2	--	--	100
2343115	ADS Lab	--	--	--	--	--	25	25	50
2343116	SQL Lab	--	--	--	--	--	25	25	50
2343611	Mini-Project - Full Stack Java Programming	--	--	--	--	--	50	25	75
2993511	Entrepreneurship Development	--	--	--	--	--	50	--	50
2993512	Environmental Science for Engineers	--	--	--	--	--	50	--	50
Total		**100**	**100**	**200**	**300**	**10**	**225**	**75**	**800**

Program Structure for Second Year of Information Technology

UNIVERSITY OF MUMBAI (With Effect from 2025-2026)

SEMESTER IV

Course Code	**Course Description**	**Teaching Scheme (Contact Hours)**			**Credit Assigned**			
		Theory	Practical	Tutorial	Theory	Tutorial	Practical	Total Credits
2344111	Applied Mathematics Thinking-II	2	--	1	2	1	–	3
2344112	Operating System	3	–	--	3	–	–	3
2344113	Computer Network & Network Design	3	--	--	3	–	–	3
MDC401	Multidisciplinary minor	3	–	--	3	–	–	3
OEC401	Open Elective	2#	–	--	2	–	–	2
2344114	Unix Lab	–	2	–	–	–	1	1
2344115	Network Design Lab	–	2	–	–	–	1	1
MDL401	Multidisciplinary minor	–	2	–	–	–	1	1
2344411	Mini-Project -Programming Paradigm	--	2*+2	--	--	--	2	2
2994511	Business Model Development	–	2*+2	–	–	–	2	2
2994512	Design Thinking	–	2*+2	–	–	–	2	2
Total		**13**	**18**	**01**	**13**	**01**	**09**	**23**

* Two hours of practical class to be conducted for full class as demo/discussion.

Theory / Tutorial 1 credit for 1 hour and Practical 1 credit for 2 hours

# Students must select course for Open Elective from Science/Commerce/Management stream bucket provided by the University of Mumbai.

#Institute shall offer a course for MDM from other Engineering Boards.

Course Code	**Course Description**	**Examination scheme**							
		Internal Assessment Test (IAT)			End Sem. Exam Marks	End Sem. Exam Duration (Hrs)	Term Work (Tw)	Oral & Pract.	Total
		IAT-I	IAT-II	Total (IAT-I) + IAT-II)					
2344111	Applied Mathematics Thinking-II	20	20	40	60	2	25	--	125
2344112	Operating System	20	20	40	60	2	--	--	100
2344113	Computer Network & Network Design	20	20	40	60	2	--	--	100
MDC401	Multidisciplinary minor	20	20	40	60	2	--	--	100
OEC401	Open Elective	20	20	40	60	2	--	--	100
2344114	Unix Lab	--	--	--	--	--	25	25	50
2344115	Network Design Lab	--	--	--	--	--	25	25	50
MDL401	Multidisciplinary minor	--	--	--	--	--	25	--	25
2344411	Mini-Project - Programming Paradigm	--	--	--	--	--	50	25	75
2994511	Business Model Development	--	--	--	--	--	50	--	50
2994512	Design Thinking	--	--	--	--	--	50	--	50
Total		**100**	**100**	**200**	**300**	**10**	**250**	**75**	**825**

Vertical – 1 Major

Sem. – III

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2343111	Applied Mathematics Thinking-I	02	-	01	02	-	01	03

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I + IAT-II					
2343111	Applied Mathematics Thinking-I	20	20	40	60	2	25	--	125

Note: * One hour of tutorial class to be conducted for full class as practice/problem solving/discussion/theory.

Course Objectives: Students will be able to learn:

1. To familiarize with the Laplace Transform, Inverse Laplace Transform of various functions, and its applications.
2. To acquaint oneself with the concept of Fourier series, its complex form, and enhance problem-solving skills.
3. To familiarize the concept of complex variables, C-R equations with applications.
4. The fundamental knowledge of Trees, Graphs, etc.
5. To study the basic techniques of statistics, including correlation, regression, and curve fitting, for data analysis, Machine learning, and AI.
6. To understand some advanced topics of probability, random variables with their Distributions and expectations.

Course Outcomes:

Sr. No.	Course Outcomes	Cognitive levels of attainment as per Bloom's Taxonomy
On successful completion, of course, learner/student will be able to:		
1	Apply the concept of Laplace transform to solve the real integrals in engineering problems.	L1, L2
2	Apply the concept of inverse Laplace transform of various functions in engineering problems.	L1, L2
3	Expand the periodic function by using Fourier series for real life problems and complex engineering problems.	L1, L2, L3
4	Find orthogonal trajectories and analytic function by using basic concepts of complex variable theory.	L1, L2, L3
5	Apply the concept of Correlation and Regression to the engineering problems in data science, machine learning and AI.	L2, L3
6	Illustrate understanding of the concepts of probability and expectation for getting the spread of the data and distribution of probabilities.	L1, L2

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
0	Prerequisite	Applied Mathematics I, Applied Mathematics-II	01	--
I	**Laplace Transform**	Definition of Laplace transform, Condition of Existence of Laplace transform,	**06**	**CO1**

		Laplace Transform of Standard Functions likee^{at}, $sin(at)$, $cos(at)$, $sinh(at)$, $cosh(at)$and t^n, $n\geq0$. Properties of Laplace Transform: Linearity, First Shifting Theorem, Change of scale Property, multiplication by *t*, Division by *t* *(division by t^2 not included),* Laplace Transform of derivatives (up to second derivative) and integrals (Properties without proof) Evaluation of real integrals by using Laplace Transformation. **Self-learning Topics:** Heaviside's Unit Step function, Second shifting theorem, Laplace Transform. Of Periodic functions, Dirac Delta Function.		
II	**Inverse Laplace Transform**	Inverse Laplace Transform, Linearity property, use of standard formulae to find inverse Laplace Transform, finding Inverse Laplace transform using derivatives, Partial fractions method to find inverse Laplace transform. (only up to s^3 in denominator) Inverse Laplace transform using Convolution theorem (without proof) Applications to solve initial and boundary value problems involving ordinary differential equations (up to 2nd order differential equation) **Self-learning Topics:** Partial fractions method to find inverse Laplace transform. (with s^4 in denominator), Inverse Laplace transform using derivatives and its properties.	**06**	**CO1, CO2**
III	**Fourier Series**	Dirichlet's conditions, Definition of Fourier series and Parseval's Identity (without proof) Fourier series of periodic function with period2□and2*l*, Fourier series of even and odd functions (simple functions only, piecewise continuous function not to be included) Hal Frange Sine and Cosine Series. **Self-learning Topics:** Complex form of Fourier Series, orthogonal and orthonormal set of functions, Fourier Transform.	**05**	**CO3**
IV	**Complex Variables**	Analytic function, necessary and sufficient conditions for *f*(z) to be analytic (without proof), Cauchy-Riemann equations in cartesian coordinates (without proof, Polar form not included) Milne-Thomson method to determine analytic function *f*(z) when real (u)or imaginary part (v) is given Harmonic function, Harmonic conjugate, and orthogonal trajectories **Self-learning Topics:** Cauchy-Riemann equations in polar coordinates, conformal mapping, linear, bilinear mapping, cross ratio, fixed points, and standard transformations.	**05**	**CO4**
V	**Statistical Techniques**	Kar lPearson's Coefficient of correlation(r) Spearman's Rank correlation coefficient (R) (with repeated and non-repeated ranks) Lines of regression Fitting of first- and second-degree curves. **Self-learning Topics:** Covariance, fitting of exponential curve.	**04**	**CO5**
VI	**Probability**	Discrete and continuous random variable with probability distribution and probability density function. Expectation of random variable with mean, variance and standard deviation, moment generating function up to two moments.	**04**	**CO6**

		Self-learning topics: Total probability theorem, Bayes' theorem, Skewness and Kurtosis of distribution (data).		
Note: No questions will be asked in the end-semester exam from self-study topics. However, students are encouraged to explore these topics for a better understanding of the subject.				

Text Books:

1. Higher Engineering Mathematics, Dr. B.S. Grewal, Khanna Publication
2. Advanced Engineering Mathematics, Erwin Kreyszig, Wiley Eastern Limited.
3. Advanced Engineering Mathematics, R.K. Jainand S.R.K. Iyengar, Narosapublication,

References:

1. Complex Variables and Applications, Brown and Churchill, McGraw-Hill Education.
2. Probability, Statistics and Random Processes, T. Veerarajan, McGraw-Hill education.
3. Theory and Problems of Fourier Analysis, Murray Spiegel, Schaum'sOutlineSeries.
4. Higher Engineering Mathematics, H. K. Dass
5. Text book of Engineering Mathematics, N. P. Bali and Dr. Manish Goyal

Online References:

Sr. No.	Website Name
1.	https://www.nptel.ac.in
2	Higher Engineering Mathematics by H.K.Dass
3	Higher Engineering Mathematics by B.V. Raamana

List Of Tutorials :

Tutorial No	Tutorial Topic	Hours
1	Laplace Transform	1
2	Inverse Laplace Transform	1
3	Application Of Laplace Transform	1
4	Fourier Series (Full range)	1
5	Half Range Fourier Series	1
6	Complex Variables	1
7	Statistical Techniques	1
8	Probability	1

Term Work:

General Instructions:

1. Students must be encouraged to write at least 6 class tutorials on the entire syllabus.
2. A group of 4-6 students should be assigned a self-learning topic. Students should prepare a presentation/problem solving of 10-15 15minutes. This should be considered as a mini project in

Applied Mathematics. This project should be graded for 10 marks depending on the performance of the students.

The distribution of Term Work marks will be as follows–

1.	Attendance (Theory and Tutorial)	05marks
2.	Class Tutorials on entire syllabus	10marks
3.	Mini project	10marks

Assessment:

Internal Assessment Test (IAT) for 20 marks each:

- IA will consist of Two Compulsory Internal Assessment Tests. Approximately 40% to 50% of the syllabus content must be covered in the IAT-I and the remaining 40% to 50% of the syllabus content must be covered in the IAT-II.

End Semester Theory Examination:

- **Question paper format**

- Question Paper will comprise a total of **six questions each carrying 15 marks Q.1** will be **compulsory** and should **cover the maximum contents of the syllabus**
- **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules)
- A total of **four questions** need to be answered.

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2343112	Advance Data Structure & Analysis	03	–	--	03	–	–	03

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I + IAT-II					
2343112	Advance Data Structure & Analysis	20	20	40	60	02	--	--	100

Course Objective: Students will able to learn:

Sr. No.	Course Objectives
1	To learn mathematical background for analysis of algorithm
2	To learn various advanced data structures.
3	To learn greed approach to solve problems.
4	To learn backtracking algorithm and maximum flow networks.
5	To learn dynamic programming methods.
6	To understand the concept of pattern matching.

Course Outcomes:

On successful completion, of course, learner/student will be able to:

Sr.No.	Course Outcomes	Cognitive Levels of Attainment as per Bloom's Taxonomy
1	Understand methods for analysis of algorithms and solve recurrence problems.	L1, L2
2	Choose appropriate advanced data structures for a given problem and calculate its complexity.	L2, L3, L4
3	Analyze the greedy programming technique to solve problems.	L2, L3, L4, L5
4	Evaluate and analyze the backtracking algorithm and understand maximum flow networks.	L2, L3, L4, L5
5	Analyze the dynamic programming technique to solve problems.	L2, L3, L4, L5
6	Select a proper pattern matching algorithm for a given problem.	L3, L4, L5

Detailed Syllabus:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mappings
0	**Prerequisite**	Overview of Data Structures: Revision of basic data structures (arrays, stacks, queues, linked lists, trees).	01	
I	**Introduction to Analysis of Algorithms**	Fundamentals of the analysis of algorithms: Time and Space complexity, Asymptotic notation, Recurrence Relations: Methods to solve recurrence relations in algorithms (Substitution, Recursion tree, Master theorem). **Self-learning Topics:** Solve problems on analysis of algorithms.	03	CO1

II	**Advanced Data Structures**	Introduction. AVL trees, B tree, B tree operations, B+ tree, Red-Black Trees, tries data structures, time complexity analysis of all problems. Graphs, Representation, Graph Traversals: Breadth First Search, Depth First Search. **Self-learning Topics:** Solve problems on AVL trees, B tree, B+ tree etc.	**08**	**CO2**
III	**Greedy algorithms and Applications**	Introduction and properties of greedy algorithms, Fractional Knapsack problem, Minimum Spanning Trees (Prim's and Kruskal's algorithms), Job sequencing with deadlines, Optimal storage on tapes, Analysis of All problems. **Self-learning Topics:** Solve problems on Spanning Trees, Knapsack etc.	**06**	**CO3**
IV	**Backtracking and Maximum flow Networks**	Backtracking Techniques: Introduction, N-Queens problem, sum of subsets problem, graph coloring, Hamiltonian cycles. Introduction to flow networks, Augmenting Paths Residual Network, Ford Fulkerson method, Applications of Flow Networks in real-world problems. **Self-learning Topics:** Solve problems N-Queens, Hamiltonian cycles, Augmenting Paths Residual Network etc.	**07**	**CO4**
V	**Dynamic Algorithms**	Introduction Dynamic algorithms, Greedy vs. Dynamic algorithms, Single source shortest path- Dijkstra's Algorithm, Bellman Ford Algorithm, All pair shortest path- Floyd Warshall Algorithm, 0/1 knapsack problem, Travelling salesman problem, Analysis of All problems. **Self-learning Topics:** Solve problems on shortest path- Dijkstra's Algorithm etc.	**08**	**CO5**
VI	**String Matching Algorithms**	Introduction. Naïve string matching algorithm, Rabin-Karp algorithm, Knuth-Morris-Pratt(KMP) algorithm, Longest common subsequence(LCS), Analysis of All problems, **Applications:** Text searching, DNA sequencing, and data compression. **Self-learning Topics:** Solve problems on DNA sequencing, and data compression.	**06**	**CO6**
Note: No questions will be asked in the end-semester exam from self-study topics. However, students are encouraged to explore these topics for a better understanding of the subject.				

Text Books and References:

Sr. No	Title	Authors	Publisher	Edition	Year
1	Introduction to Algorithms	Cormen, Leiserson, Rivest, Stein	PHI	3rd Edition	2011
2	Algorithm Design	Jon Kleinberg, Éva Tardos	Pearson	1st Edition	2006
3	Data Structures and Algorithm Analysis in C++	Mark Allen Weiss	Pearson	4th Edition	2013
4	Introduction to the Design and Analysis of Algorithms	Anany Levitin	Pearson	3rd Edition	2011
5	Algorithms	Robert Sedgewick, Kevin Wayne	Addison-Wesley	4th Edition	2011

Online Resources:

S. No.	Website Name	URL	Modules Covered
1	NPTEL	https://archive.nptel.ac.in/courses/106/106/106106200/	M1
2	NPTEL	https://archive.nptel.ac.in/courses/106/105/106105085/	M2
3	NPTEL	https://archive.nptel.ac.in/courses/106/104/106104120/	M3
4	Coursera	https://www.coursera.org/learn/algorithms-part1	M1-M3
5	MIT OpenCourseWare	https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-006-introduction-to-algorithms-fall-2011/	M1-M6

Assessment:

Internal Assessment Test (IAT) for 20 marks each:

- IA will consist of Two Compulsory Internal Assessment Tests. Approximately 40% to 50% of the syllabus content must be covered in the IAT-I and the remaining 40% to 50% of the syllabus content must be covered in the IAT-II.

End Semester Theory Examination:

- **Question paper format**

- Question Paper will comprise a total of **six questions each carrying 15 marks Q.1** will be **compulsory** and should **cover the maximum contents of the syllabus**

- **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules)

- A total of **four questions** need to be answered.

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2343113	Database Management System & Application	03	–	--	03	–	–	03

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I + IAT-II					
2343113	Database Management System & Application	20	20	40	60	02	--	--	100

Course Objective: Students will be able to learn:

Sr. No.	Course Objectives
1	To learn the basics and understand the need for database management systems for real-world applications.
2	To construct a conceptual data model for real-world applications
3	To Build a Relational Model from ER/EER.
4	To introduce the concept of SQL to store and retrieve data efficiently.
5	To demonstrate notions of normalization for database design.
6	To understand the concepts of transaction processing- concurrency control & recovery procedures.

Course Outcomes:

On successful completion, of course, learner/student will be able to:

Sr.No.	Course Outcomes	Cognitive Levels of Attainment as per Bloom's Taxonomy
1	Identify the need of Database Management System.	L1, L2
2	Design conceptual model for real life applications.	L6
3	Create Relational Model for real life applications	L6
4	Formulate query using SQL commands.	L3
5	Apply the concept of normalization to relational database design.	L3
6	Demonstrate the concept of transaction, concurrency and recovery.	L2

Detailed Syllabus:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mappings
0	**Prerequisite**	C, Python Programming.	**02**	**--**
I	**Database System Concepts, Architecture,**	Introduction, Characteristics of Databases, File system v/s Database system, Data abstraction and Data Independence, DBMS system architecture, Database Administrator (DBA), Role of DBA.	**05**	CO1

	and Applications	**Applications:** Banking Systems, E-Commerce, Telecommunications, Healthcare Systems, Social Media Platforms, Education Systems, Airline Reservation Systems, Government Applications. **Self-learning Topics:** Identify the types of Databases.		
II	**The Entity-Relationship Model**	Conceptual Modeling of a database, The Entity-Relationship (ER) Model, Entity Type, Entity Sets, Attributes and Keys, Relationship Types, Relationship Sets, Weak entity Types. Generalization, Specialization and Aggregation, Extended Entity-Relationship (EER) Model. **Self-learning Topics:** Design an ER model for any real-time case study.	**05**	CO2
III	**Relational Model & Relational Algebra**	Introduction to Relational Model, Relational Model Constraints and Relational Database Schemas, Concept of Keys: Primary Kay, Secondary key, Foreign Key, Mapping the ER and EER Model to the Relational Model, Introduction to Relational Algebra, Relational Algebra expressions for Unary Relational Operations, • Set Theory operations, • Binary Relational operation Relational Algebra Queries **Self-learning Topics:** Map the ER model designed in module II to relational schema.	**05**	CO3
IV	**Structured Query Language (SQL) & Indexing**	Overview of SQL, Data Definition Commands, Set Operations, Aggregate Function, null values, Data Manipulation commands, Data Control commands, Complex Retrieval Queries using Group By, Recursive Queries, nested Queries, Integrity constraints in SQL. Database Programming with JDBC, Security and authorization: Grant & Revoke in SQL Functions and Procedures in SQL and cursors. Indexing: Basic Concepts, Ordered Indices, Index Definition in SQL **Self-learning Topics:** Physical design of database for the relational model designed in module III and fire various queries.	**08**	CO4
V	**Relational Database Design**	Design guidelines for relational Schema, Functional Dependencies, Database tables and normalization, The need for normalization, The normalization process, Improving the design, Definition of Normal Forms- 1NF, 2NF, 3NF & The Boyce-Codd Normal Form (BCNF). **Self-learning Topics:** Consider any real-time application and normalization up to 3NF/BCNF	**07**	CO5
VI	**Transactions Management and Concurrency and Recovery**	Transaction: Transaction concept, State Diagram, ACID Properties, Transaction Control Commands, Concurrent Executions, Serializability – Conflict and View, Concurrency Control: Lock-based-protocols, Deadlock handling Timestamp-based protocols, Recovery System: Recovery Concepts, Log based recovery. **Self-learning Topics:** Study the various deadlock situations which may occur for a database designed in module V.	**07**	CO6
Note: No questions will be asked in the end-semester exam from self-study topics. However, students are encouraged to explore these topics for a better understanding of the subject.				

Text Books:

1. Korth, Slberchatz, Sudarshan, Database System Concepts, 6th Edition, McGraw Hill
2. Elmasri and Navathe, Fundamentals of Database Systems, 6th Edition, Pearson education
3. Raghu Ramkrishnan and Johannes Gehrke, Database Management Systems, TMH

References:

1. Peter Rob and Carlos Coronel, ― Database Systems Design, Implementation and Management‖, Thomson Learning, 9th Edition.
2. SQL & PL / SQL for Oracle 11g Black Book, Dreamtech Press
3. G. K. Gupta : "Database Management Systems", McGraw – Hill

Online References:

Sr. No.	Website Name
1.	https://www.nptel.ac.in
2.	https://www.oreilly.com
3.	https://www.coursera.org/

Assessment:

Internal Assessment Test (IAT) for 20 marks each:

- IA will consist of Two Compulsory Internal Assessment Tests. Approximately 40% to 50% of the syllabus content must be covered in the IAT-I and the remaining 40% to 50% of the syllabus content must be covered in the IAT-II.

End Semester Theory Examination:

➢ **Question paper format**

- Question Paper will comprise a total of **six questions each carrying 15 marks Q.1** will be **compulsory** and should **cover the maximum contents of the syllabus**

- **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules)

- A total of **four questions** need to be answered.

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2343114	Automata Theory	03	–	--	03	–	–	03

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I + IAT-II					
2343114	Automata Theory	20	20	40	60	02	--	--	100

Course Objectives: Students will able to learn:

Sr. No.	Course Objectives
1	To learn fundamentals of Regular and Context Free Grammars and Languages.
2	To understand the relation between Regular Language and Finite Automata and machines.
3	To learn how to design Automata's as Acceptors, Verifiers and Translators.
4	To understand the relation between Regular Languages, Contexts free Languages, PDA and TM.
5	To learn how to design PDA as acceptor and TM as Calculators.
6	To learn applications of Automata Theory.

Course Outcomes:

Sr. No.	Course Outcomes	Cognitive levels of attainment as per Bloom's Taxonomy
On successful completion, of course, learner/student will be able to:		
1	Explain, analyze and design Regular languages, Expression and Grammars.	L2, L4, L6
2	Design different types of Finite Automata and Machines as Acceptor, Verifier and Translator.	L6
3	Analyze and design Context Free languages and Grammars.	L4, L6
4	Design different types of Push down Automata as Simple Parser.	L6
5	Design different types of Turing Machines as Acceptor, Verifier, Translator and Basic computing machine.	L6
6	Develop understanding of applications of various Automata.	L6

Prerequisite: Basic Mathematical Fundamentals: Sets, Logic, Relations, Functions.

DETAILED SYLLABUS:

Sr. No.	Module	Detailed Content	Hours	CO Mapping
0	Prerequisite	Basic Mathematical Fundamentals: Sets, Logic, Relations, Functions.	**02**	-

I	Introduction and Regular Languages	Languages: Alphabets and Strings. Regular Languages: Regular Expressions, Regular Languages, Regular Grammars, RL and LL grammars, Closure properties **Self-learning Topics:** Practice exercise on Regular Expressions. Identify the tools also.	**05**	CO1
II	Finite Automata	Finite Automata: FA as language acceptor or verifier, NFA (with and without ε) , DFA, RE to NFA, NFA to DFA, Reduced DFA , NFA-DFA equivalence, FA to RE. Finite State Machines with output : Moore and Mealy machines. Moore and Mealy M/C conversion. Limitations of FA. **Self-learning Topics:** Practice exercise on FA and NFA	**09**	CO2
III	Context Free Grammars	Context Free Languages: CFG, Leftmost and Rightmost derivations, Ambiguity, Simplification and Normalization (CNF & GNF) and Chomsky Hierarchy (Types 0 to 3) **Self-learning Topics:** Practice numerical or exercise on CFG	**08**	CO3
IV	Push Down Automata	Push Down Automata: Deterministic (single stack) PDA, Equivalence between PDA and CFG. Power and Limitations of PDA. **Self-learning Topics:** List the examples of PDA.	**05**	CO4
V	Turing Machine	Turing Machine: Deterministic TM, Variants of TM, Halting problem, Power of TM. **Self-learning Topics:** Practice numerical of TM.	**07**	CO5
VI	Applications of Automata	Applications of FA, CFG, PDA & TM. Introduction to Compiler & Its phases. **Self-learning Topics:** Case study on any one compiler.	**03**	CO2, CO3, CO4, CO5, CO6
Note: No questions will be asked in the end-semester exam from self-study topics. However, students are encouraged to explore these topics for a better understanding of the subject.				

Text books

1. J.C. Martin, “Introduction to languages and the Theory of Computation”, TMH.
2. Kavi Mahesh, “Theory of Computation A Problem Solving Approach”, Wiley India
3. A. V. Aho, R. Shethi, Monica Lam , J.D. Ulman , “Compilers Principles, Techniques and Tools “,Pearson Education.

References

1. John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, “Introduction to Automata Theory, Languages and Computation”, Pearson Education.
2. Daniel I.A. Cohen, “Introduction to Computer Theory”, John Wiley & Sons.
3. Vivek Kulkarni,” Theory of Computation”, Oxford University.
4. N. Chandrashekhar, K.L.P. Mishra, “Theory of Computer Science, Automata Languages & Computations”, PHI publications.
5.J. J. Donovan, “ Systems Programming”, TMH.

Online References:

Sr. No.	**Website Name**
1.	https://www.nptel.ac.in
2.	https://online.stanford.edu
3.	https://www.coursera.org/

Assessment:

Internal Assessment Test (IAT) for 20 marks each:

- IA will consist of Two Compulsory Internal Assessment Tests. Approximately 40% to 50% of the syllabus content must be covered in the IAT-I and the remaining 40% to 50% of the syllabus content must be covered in the IAT-II.

End Semester Theory Examination:

- **Question paper format**

- Question Paper will comprise a total of **six questions each carrying 15 marks Q.1** will be **compulsory** and should **cover the maximum contents of the syllabus**

- **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules)

- A total of **four questions** need to be answered.

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2343115	ADSA Lab	--	2	–	–	–	1	1

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I + IAT-II					
2343115	ADSA Lab	--	--	--	--	--	25	25	50

Lab Objective: The course aims to

Sr. No	Lab Objectives
1	To learn mathematical background for analysis of algorithm
2	To learn various advanced data structures.
3	To learn greed approach to solve problems.
4	To learn backtracking algorithm and maximum flow networks.
5	To learn dynamic programming methods.
6	To understand the concept of pattern matching.

Lab Outcomes: Upon completion of the course students will be able to:

Sr. No.	Lab Outcomes	Cognitive Levels of Attainment as per Bloom's Taxonomy
1	Understand methods for analysis of algorithms and solve recurrence problems.	L1, L2
2	Choose appropriate advanced data structures for a given problem and calculate its complexity.	L2, L3, L4
3	Analyze the greedy programming technique to solve problems.	L2, L3, L4, L5
4	Evaluate and analyze the backtracking algorithm and understand maximum flow networks.	L2, L3, L4, L5
5	Analyze the dynamic programming technique to solve problems.	L2, L3, L4, L5
6	Select a proper pattern matching algorithm for a given problem.	L3, L4, L5

Hardware & Software Requirements:

Hardware Requirement:	Software requirement:
PC i3 processor and above	Turbo/Borland C complier

DETAILED SYLLABUS:

Sr. No.	Name of Module	Suggested list of Practical	Hours	LO Mappings
I	**Introduction to Analysis of Algorithms**	1. Implement Merge sort and Quicksort for the given list of integer values and find space and time complexity. 2. Implementation of randomized quicksort algorithm and find space and time complexity. 3. Implementation of hash functions and its associated algorithms.	**04**	**LO1**
II	**Advanced Data Structures**	4. Construct Binary Search Tree for given sequence of integers and perform Pre-order, In-order and Post-order traversal of constructed tree. Analyze complexities. 5. Implement Insert, Delete, Search and Display operations on Binary Search Tree and analyze space and time complexity. 6. Implementation of operations on B/B+-trees.	**05**	**LO2**
III	**Greedy algorithms and Applications**	7. Implement solution for a 0-1 knapsack problem using dynamic programming. 8. Implement Prim's and Kruskal's algorithms. 9. Implement solution for job sequencing with deadlines problems.	**05**	**LO3**
IV	**Backtracking and Maximum flow Networks**	10. Implement N-Queen's problem using Back Tracking. 11. Implement Sum of subsets problem for a given set of distinct numbers using backtracking. 12. Implement graph coloring, 13. Implement Hamiltonian cycles. 14. Implementation of Ford-Fulkerson algorithm.	**04**	**LO4**
V	**Dynamic Algorithms**	15. Implementation of Bellman-Ford algorithm. 16. Implement Floyd Warshall Algorithm. 17. Implement Travelling salesman problem.	**04**	**LO5**
VI	**String Matching Algorithms**	18. Implement Naïve string matching algorithm. 19. Implement Rabin-Karp algorithm. 20. Implement KMP algorithm. 21. Implement Longest common subsequence. 22. Implement any one application.	**04**	**LO6**

Text Books and References:

Sr. No	Title	Authors	Publisher	Edition	Year
1	Introduction to Algorithms	Cormen, Leiserson, Rivest, Stein	PHI	3rd Edition	2011
2	Algorithm Design	Jon Kleinberg, Éva Tardos	Pearson	1st Edition	2006
3	Data Structures and Algorithm Analysis in C++	Mark Allen Weiss	Pearson	4th Edition	2013

4	Introduction to the Design and Analysis of Algorithms	Anany Levitin	Pearson	3rd Edition	2011
5	Algorithms	Robert Sedgewick, Kevin Wayne	Addison-Wesley	4th Edition	2011

Online Resources:

S. No.	Website Name	URL	Modules Covered
1	NPTEL	https://archive.nptel.ac.in/courses/106/106/106106200/	M1
2	NPTEL	https://archive.nptel.ac.in/courses/106/105/106105085/	M2
3	NPTEL	https://archive.nptel.ac.in/courses/106/104/106104120/	M3
4	Coursera	https://www.coursera.org/learn/algorithms-part1	M1-M3
5	MIT OpenCourseWare	https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-006-introduction-to-algorithms-fall-2011/	M1-M6

Assessment :

Term Work: Term work shall consist of at least 10-12 practical's based on above list. Also Term work Journal must include at least 2 assignments.

Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Practical & Oral Exam: An Oral & Practical exam will be held based on Practicals.

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2343116	SQL Lab	--	2	–	–	–	1	1

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I + IAT-II					
2343116	SQL Lab	--	--	--	--	--	25	25	50

Lab Objective: The course aims to

Sr. No	Lab Objectives
1	To identify and define problem statements for real life applications.
2	To construct conceptual data model for real life applications.
3	To Build Relational Model from ER/EER and demonstrate usage of relational algebra.
4	To Apply SQL to store and retrieve data efficiently.
5	To implement database connectivity using JDBC.
6	To understand the concepts of transaction processing- concurrency control & recovery procedures.

Lab Outcomes: Upon completion of the course, students will be able to:

Sr. No.	Lab Outcomes	Cognitive Levels of Attainment as per Bloom's Taxonomy
1	Define problem statement and Construct the conceptual model for real life application.	L1, L3, L4, L6
2	Create and populate a RDBMS using SQL.	L3, L4
3	Formulate and write SQL queries for efficient information retrieval	L3, L4
4	Apply view, triggers and procedures to demonstrate specific event handling.	L1, L3, L4
5	Demonstrate database connectivity using JDBC.	L3
6	Demonstrate the concept of concurrent transactions.	L3, L4

Prerequisite: C and Python Programming.

Hardware & Software Requirements:

Hardware Requirement:	**Software requirement:**
PC i3 processor and above	Any SQL Compiler, Python/Java Programming Language

DETAILED SYLLABUS:

Sr. No.	Experiment List	Hours	LO Mapping
1.	Identify real world problem and develop the problem statement. Design an Entity-Relationship (ER) / Extended Entity-Relationship (EER) Model.	**02**	LO1
2.	Mapping ER/EER to Relational schema model.	**02**	LO1
3.	Create a database using DDL and apply integrity constraints.	**02**	LO2, LO3
4.	Perform data manipulations operations on populated database.	**02**	LO3
5.	Perform Authorization using Grant and Revoke.	**02**	LO2, LO3
6.	Implement Basic and complex SQL queries.	**02**	LO3, LO4
7.	Implementation of Views and Triggers.	**02**	LO4
8.	Demonstrate database connectivity using JDBC.	**02**	LO5
9.	Execute TCL commands.	**02**	LO4
10.	Implement functions and procedures in SQL	**02**	LO3, LO4
11.	Implementation of Cursor.	**03**	LO3, LO4
12.	Implementation and demonstration of Transaction and Concurrency control techniques using locks.	**03**	LO6

Note: Guidelines for the conduction of practical.

- **Faculty will assign one real-world case study or application to a group of 3 students, and each group is to perform the above list of experiments and then apply to their assigned case study or application.**
- **Learner must prepare a Journal of the above experiment list along with a report of their assigned case study/application.**

Text Books:

1. Korth, Slberchatz, Sudarshan, Database System Concepts, 6th Edition, McGraw Hill
2. Elmasri and Navathe, Fundamentals of Database Systems, 6th Edition, Pearson education
3. Raghu Ramkrishnan and Johannes Gehrke, Database Management Systems, TMH

References:

1. Peter Rob and Carlos Coronel, ― Database Systems Design, Implementation and Management‖, Thomson Learning, 9th Edition.
2. SQL & PL / SQL for Oracle 11g Black Book, Dreamtech Press
3. G. K. Gupta : "Database Management Systems", McGraw – Hill

Online References:

Sr. No.	Website Name
1.	https://www.nptel.ac.in
2.	https://www.oreilly.com
3.	https://www.coursera.org/

Assessment :

Term Work: Term work shall consist of 10-12 practical's based on above list. Also Term work Journal must include at report of assigned case study/application as an assignment.

Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Report of Case Study/Application) + 5 Marks (Attendance)

Practical & Oral Exam: An Oral & Practical exam will be held based on the above list of 10-12 Practicals.

Vertical – 1
Major

Sem. – IV

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2344111	Applied Mathematics Thinking-II	02	-	01	02	-	01	03

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I + IAT-II					
2344111	Applied Mathematics Thinking-II	20	20	40	60	2	25	--	125

Rationale:

Concepts in Statistics and Operation Research have wide applications in various fields of Engineering. Hence, they are included in the syllabus to help students understand how to apply them in practical applications.

Course Objectives:

The course aims:

1. To study Matrix algebra and its application in engineering problems.
2. To study Contour integrals and expansion of complex valued function in a power series.
3. To study Z-Transforms and Inverse Z-Transforms with its properties.
4. To study the concepts of probability distributions and sampling theory.
5. To study and apply Linear programming Techniques to solve the optimization Problems.
6. To study and apply Non-Linear programming Techniques to solve the optimization problems.

Course Outcomes:

Sr. No.	Course Outcomes	Cognitive levelsof attainment asper Bloom's Taxonomy
On successful completion, of course, learner/student will be able to:		
1	Apply the concepts of eigenvalues and eigen vectors to solve engineering problems.	L1, L2
2	Apply the use of concepts of Complex Integration for evaluating integrals, computing residues & evaluate various contour integrals.	L1, L2
3	Apply the concept of Z- transformation and its inverse in engineering problems.	L1, L2, L3
4	Apply the concept of probability distribution to engineering problems & testing hypothesis of small samples using sampling theory.	L1, L2, L3
5	Apply the concept of Linear Programming to solve the optimization problems.	L2, L3
6	Apply the concept of Non-Linear Programming to solve the optimization problems.	L1, L2, L3

Prerequisite: Applied Mathematics-I, Applied Mathematics-II, Applied Mathematics Thinking-I.

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
I	**Linear Algebra (Theory of Matrices)**	1.1 Characteristic Equation, Eigenvalues and Eigenvectors and properties(without proof). 1.2 Cayley-Hamilton Theorem (without proof), Reductionof higher degree polynomials. 1.3 Similarity of matrices, diagonalizable and non-diagonalizable matrices **Self-learning Topics:** Derogatory and non-derogatory matrices, Functions ofSquare Matrix, Linear Transformations, Quadratic forms.	**05**	**CO1**
II	**Complex Integration**	2.1 Cauchy's theorem, Cauchy's integral formula (without proof). 2.2 Taylor's and Laurent's series (without proof) (upto degree 2 in denominator). 2.3 Definition of Singularity, Zeroes, poles of *f(z)*, Residues, Cauchy'sResidue Theorem (without proof). **Self-learning Topics:** Application of Residue Theorem to evaluate realintegrations.	**07**	**CO2**
III	**Z Transform**	3.1 Definition and Region of convergence. Transform of standard functions: $\{k^n a^k\}$, $\{a^{\vert k \vert}\}$, $\{{}^{k+n}C_{\mathrm{n}}\, a^k\}$, $\{c^k \sin(\alpha k + \beta)\}$, $\{c^k \sinh \alpha k\}$, $\{c^k \cosh \alpha k\}$. 3.2 Properties of Z Transform: Change of Scale, Shifting Property,Multiplication, and Division by k, Convolution theorem (without proof). 3.3 Inverse Z transform: Partial Fraction Method, Convolution Method. **Self-learning Topics:** Initial value theorem, Final value theorem, Inverse of Transform by Binomial Expansion.	**07**	**CO3**
IV	**Probability Distribution and Sampling Theory**	4.1 Probability Distribution: Binomial, Poisson and Normal distribution (simple problems in finding probability only to be included). 4.2 Sampling distribution, Test of Hypothesis, Level of Significance, Critical region, One-tailed, and two-tailed test, Degree of freedom. 4.3 Test significance for Large samples. Test the significance of mean and	**08**	**CO4**

		Difference between the means of two large samples. 4.4 Students' t-distribution (Small sample). Test the significance of mean and Difference between the means of two samples. **Self-learning Topics:** Estimateparameters of a population., Chi-Square Test: Test of goodness of fit and independence of attributes, Contingency table, Yate's Correction.		
V	**Linear Programming Problems**	5.1 Types of solutions, Standard and Canonical of LPP, Basic and Feasiblesolutions, slack variables, surplus variables. 5.2 Simplex method. 5.3 Duality, Dual of LPP and Dual Simplex Method. **Self-learning Topics:** Artificial variable method, Big M method, Sensitivity Analysis, Two-Phase Simplex Method,Revised Simplex Method.	**06**	**CO5**
VI	**Nonlinear Programming Problems**	6.1 NLPP with one equality constraint (two or three variables) using themethod of Lagrange's multipliers 6.2 NLPP with one inequality constraint (two variables): Kuhn-Tucker conditions **Self-learning Topics:** Problems with two equality constraints, Problems with two inequality constraints, Unconstrained optimization: One dimensional search method (GoldenSearch method, Newton's method). Gradient Search method	**06**	**CO6**
Note: No questions will be asked in the end-semester exam from self-study topics. However, students are encouraged to explore these topics for a better understanding of the subject.				

Text Books:

1. Operations Research, Hira and Gupta, S. Chand Publication.
2. Linear Algebra, A. R. Vashishta and J. N. Sharma.
3. Fundamentals of Statistics, S. C. Gupta.
4. Advanced Engineering Mathematics, R. K. Jain and S. R. K. Iyengar, Narosa.

References:

1. Advanced Engineering Mathematics, Erwin Kreyszig, John Wiley & Sons.
2. Complex Variables and Applications, Brown and Churchill, McGraw-Hill education.
3. Probability, Statistics and Random Processes, T. Veerarajan, McGraw-Hill education.
4. Operations Research: An Introduction, Hamdy A Taha, Pearson.
5. Engineering Optimization: Theory and Practice, S.S Rao, Wiley-Blackwell.

Online References:

Sr. No.	**Website Name**
1.	https://www.nptel.ac.in

Assessment:

Internal Assessment (IA) for 20 marks:

- IA will consist of Two Compulsory Internal Assessment Tests. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test

➢ **Question paper format**

- Question Paper will comprise of a total of **six questions each carrying 15 marks. Q.1** will be **compulsory** and should **cover maximum contents of the syllabus**
- **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules)
- A total of **Four questions** needs to be answered.

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2344112	Operating System	3	–	--	3	–	–	3

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I + IAT-II					
2344112	Operating System	20	20	40	60	2	--	--	100

Course Objectives:

Students will able to learn:

1) To understand the components of Operating System &its functions.
2) To introduce the concept of a process and process management.
3) To understand basic concepts related to Inter-process Communication (IPC) like race condition, mutual exclusion, deadlock, etc. and role of an Operating System in IPC.
4) To understand the concepts and implementation of memory management policies and virtual memory.
5) To understand functions of Operating System for storage management and device management.
6) To study the need and fundamentals of special-purpose operating system with the advent of new emerging technologies.

Course Outcomes:

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

Sr. No.	Course Outcomes	Cognitive levels ofattainment as per Bloom's Taxonomy
1	Define operating systems & understand the objective of an OS & its functions.	L1, L2
2	Describe the Process, PCB & compare various process scheduling algorithms.	L1, L2, L3, L4, L5
3	Evaluate the requirement for process synchronization and coordination handled by the operating system.	L2, L3, L4, L5
4	Describe and analyze memory management, its allocation policies, and virtual memory.	L2, L3, L4, L5
5	Analyze and evaluate the services provided by the Operating System for storage management.	L2, L3, L4, L5
6	Compare the functions of various special-purpose Operating Systems.	L1, L2

Prerequisite: Programming language (C & Python).

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
0	Prerequisite	Programming Language C; Basic of Hardware i.e. ALU, RAM, ROM, HDD	**02**	--
I	**Introduction to Operating Systems**	**Basics of Operating System:** Definition, Types of Operating Systems, OS Structure and operations, Process management, Memory management, storage management. **System Structure**: Operating system services and interface, System calls and types, System boot, Operating System Design and implementation, OS structure, Virtual machines. **Self-learning Topics:** Study of any three different OS, System calls with examples for different OS	**04**	**CO1**
II	**Process Management**	**Processes:** Definition, Process states, Process State transitions, Process Control Block, Context switching, Threads, Mmultithreading, Thread models, Benefits of threads. **Process Scheduling**: Definition, Scheduling objectives, Types of Schedulers, Scheduling criteria: CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time, Scheduling algorithms: Preemptive and Non-preemptive, Thread Scheduling and Multiple Processor Scheduling. **Self-learning Topics:** Performance comparison of Scheduling Algorithms, Selection of Scheduling Algorithms for different situations, Real-time Scheduling.	**06**	**CO2**
III	**Process Synchronization**	**Synchronization**: Inter-process Communication and Synchronization; Race Condition; The Critical Section Problem, Peterson's Solution, synchronization Hardware and semaphores, Producer Consumer Problem; Message passing. **Deadlocks:** System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock. **Self-learning Topics:** Study a real time case study for Deadlock detection and recovery.	**09**	**CO3**
IV	**Memory Management**	**Memory Management strategies:** Background, Logical and Physical address map, Memory allocation: Contiguous Memory allocation, Fixed and variable partition, Internal and External fragmentation and Compaction, Swapping, Contiguous Memory Allocation, Paging, Structure of the Page Table, Segmentation; Virtual Memory basics, Hardware and control structures, Locality of reference, Demand Paging, Page replacement Algorithms, Thrashing. **Self-learning Topics:** Memory Management for any one Operating System, Implementation of Page Replacement Algorithms.	**08**	**CO4**
V	**File Management**	**File system:** File Concept, Access Methods, Directory and DiskStructure, File-System Mounting, File Sharing, Protection; Implementing file System: File-System Structure, File-System Implementation, Directory Implementation, Allocation Methods, Free-Space Management, Efficiency and Performance, Recovery. Overview of Mass Storage Structure; Disk Structure; Disk Scheduling; RAID Structure; Introduction to I/O Systems. **Self-learning Topics:** File System for Linux and Windows, Features of I/O facility for different OS.	**06**	**CO5**

VI	**Special-purpose Operating Systems**	Open-source and Proprietary Operating System; Fundamentals of Distributed Operating System; Network Operating System; Embedded Operating Systems; Cloud and IoT Operating Systems; Real-Time Operating System; Mobile Operating System; Multimedia Operating System; Comparison between Functions of various Special-purpose Operating Systems. **Self-learning Topics:** Case Study on any one Special-purpose Operating Systems.	**04**	**CO6**
Note: No questions will be asked in the end-semester exam from self-study topics. However, students are encouraged to explore these topics for a better understanding of the subject.				

Text Books:

1. A. Silberschatz, P. Galvin, G. Gagne, Operating System Concepts, 10th ed., Wiley, 2019.
2. W. Stallings, Operating Systems: Internal and Design Principles, 9th ed., Pearson, 2018.
3. A. Tanenbaum, Modern Operating Systems, Pearson, 4th ed., 2015.

References:

1. N. Chauhan, Principles of Operating Systems, 1st ed., Oxford University Press, 2014.
2. A. Tanenbaum and A. Woodhull, Operating System Design and Implementation, 3rd ed., Pearson.
3. R. Arpaci-Dusseau and A. Arpaci-Dusseau, Operating Systems: Three Easy Pieces, CreateSpace Independent Publishing Platform, 1st ed., 2018.

Online Resources:

Sr. No.	Website Name
1.	https://nptel.ac.in/courses/106/105/106105214/
2.	https://www.tutorialspoint.com/operating_system/index.htm
3.	https://swayam.gov.in/
4.	https://www.geeksforgeeks.org/operating-systems/

Assessment:

Internal Assessment (IA) for 20 marks:

- IA will consist of Two Compulsory Internal Assessment Tests. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test

➢ **Question paper format**

- Question Paper will comprise of a total of **six questions each carrying 15 marks. Q.1** will be **compulsory** and should **cover maximum contents of the syllabus**
- **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules)
- A total of **Four questions** needs to be answered.

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2344113	Computer Network & Network Design	3	--	--	3	–	–	3

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I + IAT-II					
2344113	Computer Network & Network Design	20	20	40	60	2	--	--	100

Rationale:
In today's modern world, computer networks serve as the foundation for effective data transfer, facilitating communication through email, messaging, file sharing, video calls, and streaming etc. They also allow devices to share resources such as printers, copiers, and fax machines, leading to significant cost savings. The principles of network design are relevant across various settings, from small businesses to large corporations, highlighting the versatility and importance of this knowledge. As a result, a solid understanding of computer networks is essential for anyone aspiring to work in IT, cybersecurity, or software development.

Course Objectives:

Students will be able to learn:

Sr. No.	Course Objectives
1.	Study the functionalities of each layer of the OSI and TCP/IP models.
2.	Acquire knowledge of different types of transmission media.
3.	Acquire the knowledge of data link layer concepts and their protocols for node-to-node delivery of data.
4.	Analyze the strengths and weaknesses of routing protocols and gain knowledge about IP addressing.
5.	Study the data transportation and session management issues and related protocols used for end-to-end delivery of data.
6.	Gain the knowledge of data presentation techniques used in the presentation layer & client/server model in application layer protocols.

Course Outcomes:

On successful completion, of course, the learner/student will be able to:

Sr. No.	Course Outcomes	Cognitive levels ofattainment as per Bloom's Taxonomy
CO1:	Explain the functionalities of different layers of the OSI & TCP/IP models and compare the models.	L1, L2
CO2:	Categorize the types of transmission media.	L1, L2
CO3:	Explain data link layer concepts, design issues, and protocols.	L1, L2, L3
CO4:	Analyze the network and select an appropriate routing strategy / addressing scheme to design a network for an organization.	L1, L2, L3
CO5:	Describe the mechanisms and related protocols used for end-to-end delivery of data.	L1, L2, L3
CO6:	Implement compression strategies for the application in hand and establish client-server model.	L1, L2, L3

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
0	Prerequisite	• Terminologies of communication	--	--
I	**Introduction to Computer Networks**	• Communication Model • Goals of Computer Communication Networks • Types of Computer Communication Networks • Network Topology • Layered Architecture of Data Network • OSI Reference Model • TCP/IP Reference Model • Internetworking and Network Devices	**04**	**CO1**
II	**Physical Layer**	• Guided and Unguided Transmission Media • Switching: Circuit Switching, Message Switching, Datagram Packet Switching, Virtual-Circuit Packet Switching • Structure of a switch, Space Division Switching, Time Division Switching, Packet Switch	**04**	**CO2**
III	**Data Link Layer**	*Data Link Control:* • Framing • Error Detection & Correction: basic concept, Linear Block Code, CRC code, Checksum (Simple Problems) • Flow Control: Stop-and-Wait Flow Control, Sliding Window Flow Control (Simple Problems) • Error Control: Stop-and-Wait ARQ, Go-Back-N ARQ, Selective Reject ARQ (Simple Problems) *Medium Access Control:* • Scheduled Access (Reservation, Polling, Token Passing), • Random Access (ALOHA, CSMA, CSMA/CD, CSMA/CA) (Simple Problems on throughput) *Link Layer Addressing:* • MAC address and its types • ARP *Ethernet Protocol:* • Standard Ethernet, Fast Ethernet (100 Mbps), Gigabit Ethernet, 10-Gigabit Ethernet **Self-Study Topics:** • HDLC Protocol, IEEE 802.11 Wireless LAN, Bluetooth	**09**	**CO3**
IV	**Network Layer**	*Data Transfer:* • Network Layer Services • IPv4 Addressing (Classful/Classless) (Simple Problems) • Subnetting & Supernetting (Simple Problems)	**10**	**CO4**

		• IPv4 Protocol, ICMP Protocol • IPv6 Addressing & Protocol • Transition from IPv4 to IPv6 *Routing of Packets:* • Routing Algorithms (Distance Vector Routing [Simple Problems], Link State Routing [Simple Problems], Path Vector Routing) • Routing Protocols (RIP, OSPF, BGP) *Network Design:* • Concept of VLAN, VPN • Case study on designing a network for an organization/ college **Self-Study Topics:** • NAT, IGMP, ICMPv6		
V	**Transport Layer & Session Layer**	*Transport Layer:* • Transport Layer Services, Port Number, Socket Address, Flow & Congestion Control at Transport Layer, Connectionless and Connection-Oriented Services • User Datagram Protocol (UDP): User Datagram, UDP Services, UDP Applications (Simple Problems on UDP Header) • Transmission Control Protocol (TCP): TCP Services, TCP Segment Format, TCP Timers (Simple Problems on TCP Header) *Session Layer:* • Session Layer Design Issues • Remote Procedure Call (RPC) Protocol (Handshaking) **Self-Study Topics:** • Congestion control, Quality of Service	**06**	**CO5**
VI	**Presentation Layer & Application Layer**	*Presentation Layer:* • Compression: Basics of compression, Lossless and Lossy Compression • Compression Techniques: Huffman Code, LZW Code, Run Length Code (Simple Problems) • Image Compression: GIF, JPEG *Application Layer:* • Client/Server Paradigm • Standard Applications: WWW, HTTP, FTP, Email, DNS **Self-Study Topics:** • POP, IMAP, SNMP	**06**	**CO6**

Note: No questions will be asked in the end-semester exam from self-study topics. However, students are encouraged to explore these topics for a better understanding of the subject.

Text Books:

1. Behrouz A. Forouzan, Data Communications and Networking with TCPIP Protocol Suite, 6th Edition, McGraw Hill Education, 2022.
2. Andrew S Tanenbaum, Computer Networks -, 6th Edition, Pearson Education, 2022.

References:

1. Behrouz A. Forouzan, TCP/IP Protocol Suite,4th Edition, McGraw Hill Education, 2017.
2. James F. Kurose, Keith W. Ross, Computer Networking: A Top-Down Approach, 8th Edition, Pearson, 2022.
3. Larry L. Peterson, Bruce S. Davie, Computer Networks: A Systems Approach, 6th Edition, Morgan Kaufmann, 2021.
4. Alberto Leon-Garcia, Indra Widjaja, Communication Networks: Fundamental Concepts and Key Architectures, 2th Edition, McGraw Hill Education, 2017.
5. Stallings William, Data and Computer Communications, 10th Edition, Pearson, 2017.
6. Khalid Sayood, Introduction to Data Compression, 5th Edition, Elsevier, 2019.

Online References:

1. NPTEL Course: Computer Networks and Internet Protocol, by Prof. Soumya Kanti Ghosh, Prof. Sandip Chakraborty, IIT Kharagpur. https://nptel.ac.in/courses/106105183

2. NPTEL Course: Computer Networks, by Prof. Hema A Murthy, IIT Madras. https://nptel.ac.in/courses/106106091

Assessment:

Internal Assessment (IA) for 20 marks:

- IA will consist of Two Compulsory Internal Assessment Tests. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test

➢ **Question paper format**

- Question Paper will comprise of a total of **six questions each carrying 15 marks. Q.1** will be **compulsory** and should **cover maximum contents of the syllabus**
- **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules)
- A total of **Four questions** needs to be answered.

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2344114	Unix Lab	–	2	–	–	–	1	1

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I + IAT-II					
2344114	Unix Lab	--	--	--	--	--	25	25	50

Lab Objectives:

Students will be able to learn:

Sr. No.	Lab Objectives
The Lab experiments aims:	
1	To understand architecture and installation of Unix Operating System
2	To learn Unix general purpose commands and programming in Unix editor environment
3	To understand file system management and user management commands in Unix.
4	To understand process management and memory management commands in Unix
5	To learn basic shell scripting.
6	To learn scripting using awk and perl languages.

Lab Outcomes:

Sr. No.	Lab Outcomes	Cognitive levels of attainment as per Bloom's Taxonomy
On successful completion, of course, learner/student will be able to:		
1	Understand the architecture and functioning of Unix	L1, L2
2	Identify the Unix general purpose commands	L4
3	Apply Unix commands for system administrative tasks such as file system management and user management.	L3
4	Execute Unix commands for system administrative tasks such as process management and memory management	L4
5	Implement basic shell scripts for different applications.	L3
6	Implement advanced scripts using awk & perl languages and grep, sed, etc. commands for performing various tasks.	L3

Prerequisite: Programming Language C

Hardware & Software Requirements:

Hardware Requirement:	Software requirement:
PC i3 processor and above	Unix, Editor, Bash shell, Bourne shell and C shell

DETAILED SYLLABUS:

Sr. No.	Module	Detailed Content	Hours	LO Mapping
0	Prerequisite	Basic Programming Skills, Concepts of Operating	02	-

		System		
I	Introduction to Unix	Case Study: Brief History of UNIX, Unix Architecture; Installation of Unix Operating System	**03**	LO1
II	Basic Commands	a) Execution of Unix General Purpose Utility Commands like echo, clear, exit, date, time, uptime, cal, cat, tty, man, which, history, id, pwd, whoami, ping, ifconfig, pr, lp, lpr, lpstat, lpq, lprm, cancel, mail, etc. b) Working with Editor Vi/other editor.	**03**	LO2
III	Commands for File System Management and User Management	a) Study of Unix file system (tree structure), file and directory permissions, single and multiuser environment. b) Execution of File System Management Commands like ls, cd, pwd, cat, mkdir, rmdir, rm, cp, mv, chmod, wc, piping and redirection, grep, tr, echo, sort, head, tail, diff, comm, less, more, file, type, wc, split, cmp, tar, find, vim, gzip, bzip2, unzip, locate, etc. c) Execution of User Management Commands like who, whoami, su, sudo, login, logout, exit, passwd, useradd/adduser, usermod, userdel, groupadd, groupmod, groupdel, gpasswd, chown, chage, chgrp, chfn, etc.	**04**	LO3
IV	Commands for Process Management and Memory Management	a) Execution of Process Management Commands like ps, pstree, nice, kill, pkill, killall, xkill, fg, bg, pgrep, renice, etc. b) Execution of Memory Management Commands like free, /proc/meminfo, top, htop, df, du, vmstat, demidecode, sar, pagesize, etc.	**04**	LO4
V	Basic Scripts	a) Study of Shell, Types of Shell, Variables andOperators b) Execute the following Scripts (at least 6): (i) Write a shell script to perform arithmetic operations. (ii) Write a shell script to calculate simple interest. (iii) Write a shell script to determine largest among three integer numbers. (iv) Write a shell script to determine a given year is leap year or not. (v) Write a shell script to print multiplication table of given number using while statement. (vi) Write a shell script to search whether element is present is in the list or not. (vii) Write a shell script to compare two strings. (viii) Write a shell script to read and check if the directory / file exists or not, if not make the directory / file. (ix) Write a shell script to implement menu-driven calculator using case statement. (x) Write a shell script to print following pattern: * * * * * *	**04**	L02, L03, L05

		* * * * (xi) Write a shell script to perform operations on directory like: display name of current directory; display list of directory contents; create another directory, write contents on that and copy it to a suitable location in your home directory; etc.		
VI	Advanced Scripts	a) Execute the following scripts using grep / sed commands: (i) Write a script using grep command to find the number of words character, words and lines in a file. (ii) Write ascriptusing egrep command to display list of specific type of files in the directory. (iii) Write a script using sed command to replace all occurrences of particular word in given a file. (iv) Write a script using sedcommand to print duplicated lines in input. b) Execute the following scripts using awk / perl languages: (i) Write an awk script to print all even numbers in a given range. (ii) Write an awk script to develop a Fibonacci series (take user input for number of terms). (iii) Write a perl script to sort elements of an array. (iv) Write a perl script to check a number is prime or not.	**06**	LO2, L03, L06

Text Books:

1. S. Das, Unix Concepts and Applications, 4th ed., McGraw Hill, 2017.
2. R. Michael, Mastering Unix Shell Scripting, 2nd ed., Wiley, 2008.
3. D. Ambawade, D. Shah, Linux Labs and Open Source Technologies, Dreamtech Press, 2014.

References:

1. Y. Kanetkar, Unix Shell Programming, BPB Publications, 2003.
2. B. Forouzan and R. Gilberg, Unix and Shell Programming, Cengage Learning, 2003.

Assessment :
Term Work: Term work shall consist of at least 10-12 practical’s based on above list. Also Term work Journal must include at least 2 assignments.

Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Practical & Oral Exam: An Oral & Practical exam will be held based on Practicals.

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2344115	Network Design Lab	–	2	–	–	–	1	1

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I + IAT-II					
2344115	Network Design Lab	--	--	--	--	--	25	25	50

Lab Objectives:

Sr. No.	Lab Objectives
The Lab experiments aims:	
1	To get familiar with the basic network administration commands
2	To install and configure network simulator and learn basics of TCL scripting.
3	To understand the network simulator environment and visualize a network topology and observe its performance
4	To implement client-server socket programs.
5	To observe and study the traffic flow and the contents of protocol frames.
6	To design and configure a network for an organization

Lab Outcomes:

Sr. No.	Lab Outcomes	Cognitive levels of attainment as per Bloom's Taxonomy
On successful completion, of course, the learner/student will be able to:		
1	Execute and evaluate network administration commands and demonstrate their use in different network scenarios	L3, L5
2	Demonstrate the installation and configuration of the network simulator.	L1, L2
3	Demonstrate and measure different network scenarios and their performance behavior.	L1, L2
4	Implement the socket programming for client-server architecture.	L3
5	Analyze the traffic flow of in TCP/IP protocols.	L4
6	Design a network for an organization using a network design tool.	L6

Prerequisite: C /Python programming. Basic commands of Windows and Linux operating systems.

Hardware & Software Requirements:

Hardware Requirement:	Software requirement:
PC i3 processor and above	NS2.34 or higher version, Protocol Analyzer (e.g., Wireshark), C/Java/python

DETAILED SYLLABUS:

Sr. No.	Module	Detailed Content	Hours	LO Mapping
I	**Fundamentals of Computer Network**	• To study basic networking commands in windows operating system- Understanding Basic networking Commands: ifconfig ,ip, traceroute, tracepath, ping, netstat, ss, dig, nslookup, route, host, arp, hostname, curl or wget, mtr, whois, tcpdump • Execute and analyze basic networking commands. • To study basic networking commands in Unix/Linux operating system	**02**	**LO1**
II	**Network Simulation Software**	• Installation and configuring of NS-2 simulator- Introduction to Tcl Hello Programming. • Installation and configuring of Cisco Packet Tracer.	**02**	**LO2**
III	**Simulation of Network Topologies and Protocols**	• To implement number of nodes and physical layer configuration using NS2. • To implement the given network topology and transmit data over the shared links using NS2. • Implement distance vector and link state routing protocols in NS2. • To Simulate and study stop and Wait protocol using NS2. • To Simulate Sliding Window protocol using NS2. • To configure and compare different network topologies using Cisco Packet Tracer. • To configure static routes in a network using Cisco Packet Tracer. • Performing dynamic routing in a network using Cisco Packet Tracer. • To perform subnetting/supernetting using Cisco Packet Tracer. • To configure DNS, DHCP, FTP, SMTP server (any one) on Cisco Packet Tracer. • To create a VLAN using Cisco Packet Tracer.	**10**	**LO3**
IV	**Socket Programming**	• To study and implement Socket Programming using TCP. • To study and implement Socket Programming using UDP.	**04**	**LO4**
V	**Protocol Analyzer**	• Install one of the Network protocol analyzer tools like Wireshark, tcpdump, Windump, Microsoft Message Analyzer, Ettercap, Nirsoft SmartSniff etc. • To simulate TCP/IP stack using Network protocol analyzer and analyze the network traffic.	**04**	**LO5**
VI	**Network Design**	• Design a network for an organization using the concepts of Addressing (IP Address Assignment), Naming (DNS) and Routing. • Case Study: Study the network of the institute to identify/understand transmission media, connectors, networking devises, addressing scheme, security features used.	**04**	**LO6**

Text Books:

1. Computer Network Simulation in NS2 Basic Concepts and Protocol Implementation.-Prof Neeraj Bhargava, Pramod Singh Rathore, Dr. Ritu Bhargava, Dr. Abhishek Kumar, First Edition. BPB Publication.
2. Packet analysis with Wire shark, Anish Nath, PACKT publishing
3. TCP/IP Protocol Suite 4th Edition by Behrouz A. Forouzan

References:

1. The Network Simulator ns-2: Documentation: https://www.isi.edu/websites/nsnam/ns/ns-documentation.html
2. Practical Packet Analysis: Using Wireshark to Solve Real-World Network Problems by Chris Sanders
3. Cisco Packet Tracer: https://www.netacad.com/cisco-packet-tracer

Assessment :
Term Work: Term work shall consist of at least 10-12 practical's based on above list. Also Term work Journal must include at least 2 assignments.

Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Practical & Oral Exam: An Oral & Practical exam will be held based on Practicals.

Vertical – 4
VSEC

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2344411	Mini-Project - Programming Paradigm	--	2*+2	–	–	–	2	2

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I + IAT-II					
2344411	Mini-Project - Programming Paradigm	--	--	--	--	--	50	25	75

Lab Objectives:

1. **Understand** the fundamental concepts of different programming paradigms—including Imperative, Declarative, Procedural, Object-Oriented, Functional, Event-Driven, and Logic programming—and **explain** core language design elements such as names, scopes, bindings, type systems, and memory management.
2. **Analyze** the strengths, limitations, and best practices of each programming paradigm, **evaluating** their suitability for solving a diverse range of computational problems.
3. **Apply** the most appropriate programming paradigms to **develop** efficient software solutions, tailoring these solutions to specific problem domains while ensuring performance, modularity, and scalability.
4. **Integrate** object-oriented programming principles, functional programming techniques, and concurrent programming strategies to **design** real-world applications that are robust, maintainable, and efficient.
5. **Implement** logic programming techniques to **solve** declarative problem-solving tasks in areas such as knowledge-based systems, automated reasoning, and artificial intelligence, specifically using languages like Prolog.
6. **Create** scalable and high-performance software architectures by **architecting** solutions that leverage modern design patterns, concurrency models, and advanced programming techniques aligned with real-world needs.

Lab Outcomes:

1. Students will **identify** and **compare** various programming paradigms and their core language design concepts, including names, scopes, bindings, and type systems.
2. Students will **develop and implement** imperative and procedural programming solutions, demonstrating control flow, structured programming, and parameter-passing techniques in languages such as C and Dart.
3. Students will **apply** object-oriented principles, including inheritance, polymorphism, and encapsulation, to **design** software solutions. They will also **analyze** the use of advanced OOP techniques such as interfaces, abstract classes, and exception handling in Java and C++.
4. Students will **understand** functional programming concepts such as pure functions, lambda calculus, and higher-order functions, and **apply** them to design declarative programs in Haskell.
5. Students will **evaluate** logic programming paradigms and **apply** them to solve complex declarative problems, using Prolog to create rules, facts, and queries, with a focus on goal-oriented execution and resolution strategies.

6. **Apply and Create:** Students will **apply** concurrency concepts, including multithreading and synchronization, and **create** event-driven systems using languages like Java and Dart. They will also **develop** scalable solutions to avoid race conditions, deadlocks, and ensure high performance.

Hardware & Software Requirements:

Prerequisite:

DETAILED SYLLABUS

Sr. No.	Module	Detailed Content	Hours	CO Mapping
0	**Prerequisite**	**Compilation and interpretation, Focus on overview of compilation steps.**	**1**	–
1	**Introduction to Programming Paradigms and Core Language Design Concepts**	● Introduction to different programming paradigms. **Key Paradigms**: Imperative, Declarative, Procedural, Object-Oriented, Functional, Event-Driven, and Logic programming ● Names, Scopes, and Bindings, Scope Rules, Storage Management. ● Type Systems, Type Checking, Equality Testing and Assignment. **Self Study Topics**: Study of different programming styles in Dart, Python, and Scala	5	CO 1
2	**Imperative and Procedural Paradigm**	● **Imperative Programming:** Core concepts like variables, data types, control flow abstractions (loops, conditionals). ● **Procedural Programming:** Structuring code using procedures, subroutines, and functions. Parameter passing methods - pass by value, pass by address, pass by reference. ● **Structured Programming:** Emphasizing modularity, code readability, and scope management. **Languages**: C, Dart **Self-Study Topics:** Exploring programming implementations to understand the use of Generic subroutines, Modules, Coroutines and Event handling	4	CO 2

3	**Object-Oriented Paradigm**	● **Core OOP Concepts**: Abstraction, Encapsulation, Inheritance, Polymorphism, Dynamic binding and Object Lifecycle. ● **Advanced OOP Techniques**: Interfaces, Abstract Classes. ● **Exception Handling** **Languages**: Java, C++ **Self-Study Topics:** Analyzing and designing applications using OOP principles.	5	CO 3
4	**Functional Programming Paradigm**	● Introduction to Lambda Calculus ● **Functional Programming features**- Pure Functions, functions as first class members, Curried Functions, Basic and Composite Types and Type Classes ● Pattern Matching, Guard Expressions, Evaluation Order, Higher Order Functions **Language: Haskell** **Self-Study Topics:** Identifying functional features in modern programming languages like Dart, Python.	4	CO 4
5	**Logic Programming Paradigm**	● **Logic Programming Basics:** Declarative programming, facts, rules, and queries. ● **Prolog Basics:** Syntax, unification, backtracking, and logical inference. ● **Resolution and Proof Search:** Goal-oriented execution and resolution strategies. **Languages**: **Prolog** **Self-Learning Topics:** Use of logic programming for expert systems, automated reasoning, and AI.	4	CO 5
6	**Concurrent and Event-Driven Paradigms**	● **Concurrency Concepts**: Multithreading, parallelism, synchronization, and avoiding race conditions and deadlocks. ● **Event-Driven Programming**: Event loops, callbacks, asynchronous programming, and handling asynchronous events.	6	CO 6

		Languages: Java (Concurrency model), Dart (Limited to study of event driven programming pattern)		

Textbooks

1	Scott M L, Programming Language Pragmatics, 3rd Edn., Morgan Kaufmann Publishers, 2009
2	Graham Hutton, Programming in Haskell, 2nd Edition, Cambridge University Press, 2016
3	Ravi Sethi,, Programming Languages: Concepts and Constructs; 2nd Edition, Pearson Education Asia, 1996.
4	Jonathan Sande, Kodeco Tutorial Team, Dart Apprentice: Beyond the Basics (First Edition): Object-Oriented Programming, Concurrency & More: 1st edition, Kodeco Incorporated, 2022, 2022

Reference Books

1	Programming Languages: Design and Implementation (4th Edition), by Terrence W. Pratt, Marvin V. Zelkowitz, Pearson, 2000
2	Rajkumar Buyya, Object-oriented Programming with Java: Essentials and Applications, Tata McGraw Hill Education Private Limited
3	Max Bramer, Logic Programming with Prolog, Springer ISBN-13: 978-1852-33938-8

Online Resources

Sr	Module	Free Online Courses
1	Module I	https://see.stanford.edu/Course/CS107 https://ocw.mit.edu/collections/introductory-programming/ https://see.stanford.edu/Course/CS106A Principles of programming Languages, https://nptel.ac.in/courses/106102067
2	Module II	
3	Module III	
4	Module IV	Learn You Haskell For Great Good, https://learnyouahaskell.com/chapters https://www.coursera.org/learn/scala-functional-programming
5	Module V	Online Prolog Learning Resources, https://swish.swi-prolog.org/example/examples.swinb https://www.coursera.org/learn/logic-introduction https://www.coursera.org/learn/an-introduction-to-logic-in-computer-science https://ocw.mit.edu/courses/6-001-structure-and-interpretation-of-computer-programs-spring-2005/resources/8a-logic-programming-part-1/
6	Module VI	Book Companion Resource : Dart Apprentice: Beyond the Basics (First Edition): Object-Oriented Programming, Concurrency & More https://github.com/kodecocodes/dabb-materials/tree/editions/1.0 https://see.stanford.edu/Course/CS107 https://ocw.mit.edu/courses/6-001-structure-and-interpretation-of-computer-programs-spring-2005/resources/8a-logic-programming-part-1/

Guidelines for Labs Capstone Mini-Project:

Suggested List of Experiments

Sr. No.	Module	Detailed Content	Hours	LO Mapping
0	Prerequisite	Demonstrate Compilation and interpretation stages to students for C, C++, JAVA along with how to debug the code.	02	–
I	Object Oriented Programming	**At least Two** Programming Implementations Preferably in C++ to demonstrate concepts like - Abstraction & Encapsulation, Initialization and Finalization, Inheritance, Polymorphism and Dynamic Binding.	06	LO 1
II	Run Time Program Management	**At least Two** Programs to understand Exception handling and Garbage collection, preferably in JAVA. Students should understand checked and unchecked exceptions as well as using multiple catch blocks	02	LO 2
III	Concurrent Programming	**At least Two** Program preferably in Java/C++ to demonstrate the Thread management and Synchronization	02	LO 3
IV	Functional Programming	**Tutorial on** Introduction to Haskell programming environment and Basic operators, types, prelude library functions, list and tuples in Haskell **At least Four** Haskell Programs to demonstrate Functional Programming Concepts. Sample Programs but not limited to: ● Implement a safetail function that behaves in the same way as tail, except that safetail maps the empty list to the empty list, whereas tail gives an error in this case. Define safetail using: (a) a conditional expression; (b) guarded equations; (c) pattern matching. ● Simple List Comprehension ● Higher-Order Functions ● Write a recursive function to multiply two natural numbers that use a predefined add function. ● Haskell code to represent infinite list e.g. fibonacci series	06	LO 4

		● Implement simple Calculator Students should clearly understand the syntax and the execution of the Functional Implementation using Haskell.		
V	Logic Programming	**Tutorial** on working of SWI Prolog Environment and basic understanding of facts and rules **Implement at least 2** Prolog programs to understand declarative programming concepts. The programs must be based on creating a Knowledge Base having multiple facts and rules. Students must be able to learn to query it and understand query execution using the backward chaining.	06	LO 5
VI	Concurrency and Parallelism through Dart Programming	Implement any 3 sample programs in Dart based on learning implementation of concurrency, Streams and Isolates	06	LO 6

Guidelines for Capstone Mini-Project

- Students shall form a group of 3 to 4 students, and a group shall not be allowed less than three or more than four students, as it is a group activity.
- Students should do the survey and identify needs, which shall be converted into a problem statement for the mini project in consultation with the faculty supervisor/head of department/internal committee of faculty.
- Students' hall submits an implementation plan in the form of a Gantt/PERT/CPM chart, which will cover the weekly activity of a mini-project.
- A log book to be prepared by each group, wherein the group can record weekly work progress, and the guide/supervisor can verify and record notes/comments.
- Faculty supervisor may give inputs to students during the mini project activity; however, focus shall be on self-learning.
- Students in a group shall understand the problem effectively, propose multiple solutions, and select the best possible solution in consultation with the guide/ supervisor.
- Students shall convert the best solution into a working model using various components of their domain areas and demonstrate.
- The solution to be validated with proper justification and a report to be compiled in a standard format of the University of Mumbai.

Guidelines for Assessment of Capstone Mini-Project:

Term Work

- The review/ progress monitoring committee shall be constituted by the head of departments of each institute. The progress of the mini project to be evaluated on continuous basis, minimum two reviews in each semester.
- In continuous assessment, focus shall also be on each individual student, assessment based on the individual's contribution in group activity, their understanding, and response to questions.
- Distribution of Term work marks for both semesters shall be as below;
  - Marks awarded by guide/supervisor based on above practical list and assignment completed by the students : 10

- Marks awarded by guide/supervisor for Capstone Project Completion: :10
- Quality of Project report & Project Review : 05

Review/progress monitoring committee may consider following points for assessment based of the semester project as mentioned in general guidelines.

- In this case in students' group shall complete project in all aspects including,
  - Identification of need/problem
  - Proposed final solution
  - Procurement of components/systems
  - Building prototype and testing
- Two reviews will be conducted for continuous assessment,
  - First shall be for the finalisation of problem and proposed solution
  - Second shall be for the implementation, testing and validation of solution.

Assessment criteria of Capstone Mini-Project.

Capstone Mini-Project shall be assessed based on following criteria;

1. Quality of survey/ need identification
2. Clarity of Problem definition based on need.
3. Innovativeness in solutions/ Novelty in solutions.
4. Feasibility of proposed problem solutions and selection of best solution
5. Collection of Dataset.
6. Cost effectiveness
7. Societal impact
8. Innovativeness
9. Cost effectiveness and Societal impact
10. Full functioning of working model as per stated requirements
11. Effective use of skill sets
12. Effective use of standard engineering norms
13. Contribution of an individual's as member or leader
14. Clarity in written and oral communication

Guidelines for Assessment of Capstone Mini-Project Practical/Oral Examination:

- Report should be prepared as per the guidelines issued by the University of Mumbai.
- Mini Project shall be assessed through a presentation and demonstration of working model by the student project group to a panel of Internal and External Examiners preferably from industry or research organisations having experience of more than five years approved by head of Institution.
- Students shall be motivated to publish a paper based on the work in Conferences/students competitions.

List of Sample Capstone Mini-Project :

1. Object-Oriented Programming (OOP)

a. Library Management System

- Use classes for Books, Users, Transactions.
- Implement inheritance for different types of users (students, staff).
- Optional: Add GUI using Tkinter or JavaFX.

b. Online Food Ordering System (Console-based)

- Classes: Customer, MenuItem, Order, Restaurant
- OOP features like encapsulation, polymorphism.

c. Expense Tracker App (Java or Python)

- Use OOP to model Users, Transactions, and Categories.

2. Functional Programming

a. Weather Data Analyzer (Haskell / Python with functools)

- Read weather data and apply pure functions for analysis.
- Avoid mutable data.
- Use map, reduce, filter extensively.

b. To-Do List Manager (Scala or Elixir)

- Immutable data structures
- Recursion instead of loops
- No side effects in logic layer.

3. Procedural Programming

a. Student Grade Calculator (C / Python)

- Use functions and data structures to store and compute data.
- No OOP involved.

b. ATM Simulator

- Menu-based system using functions only (no classes).
- PIN check, balance enquiry, cash withdrawal, etc.

4. Logic Programming

a. Family Tree Generator (Prolog)

- Define relations like parent, sibling, cousin using rules.
- Query relationships dynamically.

b. Sudoku Solver (Prolog)

- Define rules and constraints to solve 9x9 Sudoku puzzle.

5. Multi-Paradigm Approach

a. Chatbot using Python (OOP + Functional)

- OOP for structuring code (User, Bot, ChatSession).
- Functional for processing input (filters, parsers).

b. File Organizer Script

- Script to organize files into folders based on type.
- Use procedural logic + OOP for file operations.

6. Event-Driven Programming

a. GUI Calculator (Python Tkinter / Java Swing)

- Event listeners for button presses.
- Follow MVC pattern optionally.

b. Traffic Light Simulation (JavaScript + HTML/CSS)

- Events for light changes
- Use state variables to track behavior

Assessment :

Term Work: Term Work shall consist of list of all practicals' based on the above list. Also, the Term work Journal must include at least 2 assignments and Mini-Project Report.

Term Work Marks: 50 Marks (Total marks) = 10 Marks (Experiment) + 5 Marks (Assignments) + 30 Marks (Capstone Mini- Project with full prototype/ product demo, testing, validation and Report) + 5 Marks (Attendance).

Oral Exam: An Oral exam will be held based on the Capstone Mini-Project.

Vertical – 6 CEP

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2343611	Mini-Project - Full Stack Java Programming	–	2*+2	–	–	–	2	2

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I + IAT-II					
2343611	Mini-Project - Full Stack Java Programming	--	--	--	--	--	50	25	75

Lab Objectives:

Sr. No.	Lab Objectives
The Lab experiments aims:	
1	To set up development environments for Java full-stack projects.
2	To develop a web interface using front-end technologies.
3	To build RESTful web services using Spring Boot.
4	To implement database CRUD operations using Hibernate/JPA.
5	To integrate front-end and back-end applications.
6	To deploy the application and present the capstone project.

Lab Outcomes:

Sr. No.	Lab Outcomes	Cognitive levels of attainment as per Bloom's Taxonomy
On successful completion, of course, the learner/student will be able to:		
1	Setup and configure full-stack Java development environment.	L1, L2
2	Develop responsive user interfaces with HTML, CSS, and React.js.	L1, L2, L3
3	Create RESTful APIs using Spring Boot framework.	L1, L2, L3
4	Perform CRUD operations with MySQL database integration.	L1, L2, L3, L4
5	Integrate front-end and back-end with secure API endpoints.	L1, L2, L3, L4
6	Deploy full-stack application on cloud platform and present capstone.	L1, L2, L3, L4, L5, L6

Prerequisite: C /Python programming. Basic commands of Windows and Linux operating systems.

Hardware & Software Requirements:

Hardware Requirement:	**Software requirement:**
PC i3 processor and above. 8GB RAM (minimum), 16GB recommended, 256GB SSD recommended, Stable internet connection.	JDK 17+, IntelliJ IDEA / Eclipse, VS Code, MySQL / MongoDB, Postman, Git, GitHub, Spring Boot, Angular CLI., Node.js & npm (for frontend React.js)AWS / Heroku Account for Deployment

DETAILED SYLLABUS:

Sr. No.	Module	Detailed Content	Hours	LO Mapping
I	**Introduction to Full Stack Java**	Overview of full-stack development: Definition of Full Stack Development, Components of Full Stack: Frontend, Backend, Database, DevOps, Full Stack Developer Roles & Skills, Technology Stack Overview: Frontend: HTML, CSS, JavaScript, React.js, Backend: Java, Spring Boot, Database: MySQL, Tools: Git, Docker, Postman, Real-world applications of full stack, Java SE concepts: JVM, JDK, JRE, Data types, Variables, Operators, Control Statements, Introduction to OOP- Basic of OOP, Packages and Import statements, Exception Handling Basics, Introduction to Java EE- Need for Java EE in enterprise applications, Multi-tier architecture, MVC Architecture- Model: Business logic, Data management, View: User interface, Controller: Request handling, linking model and view. **Self-Learning Topics:** Application Servers vs. Web Servers, Java EE vs. Spring Boot: Modern development trends. MVC in Java Web Development.	**04**	**LO1**
II	**HTML 5, CSS 3 with Bootstrap**	**HTML Basics & Structure**: Learn HTML5 structure, semantic elements, character effects, document spacing, and working with tables, lists, and hyperlinks. **Responsive Web Design**: Learn to build websites that adapt to different screen sizes and devices using media queries, flexible layouts, and mobile-first design principles. **Image Handling & Forms**: Understand image roles, adding images to web pages, using images as links, and handling user input through various form elements (text fields, checkboxes, radio buttons, etc.). **CSS3 Fundamentals**: Explore CSS syntax, selectors, text formatting, fonts, colors, borders, and advanced features like CSS Grid. **Bootstrap Introduction & Components**: Get started with Bootstrap, utilizing grid systems and essential components like headers, dropdowns, and navigation bars.React.js basics, Fetch API, Axios for API calls. **Self-Learning Topics:** JavaScript DOM Manipulation, CSS Flexbox, Version Control with Git.	**04**	**LO2**
III	**Back-end development with Java**	Spring Boot introduction- What is Spring Framework? Why Spring Boot? Features of Spring Boot: Auto-configuration, Starter dependencies, embedded server, Spring Initializr, Project structure overview: pom.xml, application. Properties, Running your first Spring Boot application, RESTful API development- What is REST? REST vs. SOAP,HTTP Methods: GET, POST, PUT, DELETE, RESTful API Design principles, Creating REST Controllers in Spring Boot, Mapping HTTP methods to controller methods, Understanding Request Mapping, Get Mapping, Post Mapping, etc., Controllers and Services- Role of Controller in MVC, Defining Controllers in Spring Boot, @RestController vs. @Controller, Service layer responsibilities, Creating Service classes and injecting them	**05**	**LO3**

		using @Autowired, Dependency Injection concepts, Repositories- Introduction to Spring Data JPA, Configuring database connection in Spring Boot (H2/MySQL), Entity classes and @Entity annotation, JpaRepository and CrudRepository interfaces, Basic CRUD operations: save(), findById(), findAll(), deleteById(), Exception Handling- Importance of exception handling, @ExceptionHandler annotation, Creating global exception handlers with @ControllerAdvice, Custom exception classes, Standard error responses (HTTP status codes, error messages).. **Self-Learning Topics:** Advanced Techniques of REST API and JPA.		
IV	**Database Integration**	Introduction to MySQL- What is a relational database? Introduction to RDBMS, Basics of MySQL: Database, Tables, Records, Primary & Foreign Keys, MySQL Workbench or command-line interface, Database design basics: ER diagrams, normalization, Setting up MySQL: installation, creating databases, tables, Introduction to SQL: Data Definition Language (DDL): CREATE, ALTER, DROP, Data Manipulation Language (DML): INSERT, UPDATE, DELETE, SELECT, CRUD operations- What is CRUD: Create, Read, Update, Delete, Writing SQL queries for CRUD operations: INSERT INTO table, SELECT FROM table, UPDATE table SET, DELETE FROM table, Filtering and Sorting data: WHERE, ORDER BY, LIKE, LIMIT, Using Aggregate Functions: COUNT, AVG, SUM, MAX, MIN, Joins: INNER JOIN, LEFT JOIN, JPA & Hibernate ORM, Repository interfaces, Query methods- Introduction to JpaRepository and Crud Repository interfaces, Default CRUD methods: save(), findById(), findAll(), deleteById(), Creating custom query methods using method naming conventions: findByName(), findByAgeGreaterThan(), etc.Pagination and Sorting with Spring Data JPA, Introduction to @Query annotation for custom JPQL/SQL queries, Native queries vs. JPQL.. **Self-Learning Topics:** JPQL Advanced Queries, Transactions in JPA, Optimizing JPA Performance.	**04**	**LO4**
V	**Full Stack Integration**	Connecting React frontend with Spring Boot backend- Overview of frontend-backend communication in full-stack applications, Setting up React app (Vite/CRA), Axios or Fetch API for making HTTP requests, Configuring CORS in Spring Boot for cross-origin access, Calling Spring Boot REST API from React components, Handling API responses in React (state management with useState/useEffect), API testing (Postman)- Introduction to Postman and its features, Creating API collections in Postman, Sending GET, POST, PUT, DELETE requests, Testing APIs with different payloads, Setting up environment variables in Postman, Automating tests with Postman scripts (basics), Error handling- Backend: Handling exceptions using @ControllerAdvice and @ExceptionHandler, Sending structured error responses (HTTP status codes, error messages, timestamps), Frontend: Handling API errors in React using try-catch and Axios interceptors, Displaying error messages to users in the React UI, Logging errors for debugging and maintenance, JWT for authentication- What is JWT? Structure of JWT: Header, Payload, Signature, Stateless authentication vs. session-based, Generating JWTs in Spring Boot, Validating and parsing JWTs, Security configuration using Spring Security, Storing JWT securely (localStorage/sessionStorage), Securing Frontend & Backend with JWT- Adding Authorization header in API requests from React, Protecting backend endpoints using JWT authorization, Role-based access control (RBAC) basics, Frontend: Securing routes and redirecting unauthenticated users, Refresh token strategy (basic	**05**	**LO4, LO5**

		concept). **Self-Learning Topics:** Students can secure full-stack applications with JWT, protect API endpoints, and manage authentication flow.		
VI	**Deployment & Capstone Project**	Docker basics- What is Docker? Containers vs. Virtual Machines, Docker architecture: Images, Containers, Docker Daemon, Docker Hub, Setting up Docker environment, Writing Dockerfile: Basic syntax, Creating images for Java and Node.js applications, Docker commands:`docker build`, `docker run`, `docker ps`, `docker stop`, `docker rm`, `docker exec`, Managing Docker images and containers, Overview of docker-compose (introductory level), Deploying Spring Boot application- Creating Dockerfile for Spring Boot, Multi-stage builds for optimized Docker images, Exposing application port using `EXPOSE`, Environment variables and configuration management, Building and running Spring Boot Docker container, Accessing REST API from a containerized Spring Boot app, Hosting React application- Building React app for production (`npm run build`), Serving static files using NGINX or simple Node.js server, Writing Dockerfile for React application, Connecting React container to Spring Boot container, Basics of Docker networking (`docker network create`), Running both containers simultaneously using docker-compose, Capstone project implementation & presentation- Finalizing capstone project scope: features, architecture, roles, Implementing core functionalities (CRUD operations, authentication, API integration), Containerizing and deploying complete full-stack application, Preparing project documentation and presentation slides, Presenting project flow: architecture diagram, tech stack, demo. **Self-Learning Topics:** Full deployment with advanced techniques.	**04**	**LO6**

Textbooks:

1. "Spring in Action" by Craig Walls
2. "Pro Spring Boot 3" by Felipe Gutierrez
3. "React Up & Running" by Stoyan Stefanov
4. "Java: The Complete Reference" by Herbert Schildt

Reference Books:

1. "Full Stack Development with Spring Boot and React" by Juha Hinkula
2. "Building Java Programs" by Stuart Reges, Marty Stepp

Online Resources:

Sr. No.	Website Name
1.	Spring Boot Official Documentation — https://spring.io/projects/spring-boot
2.	React.js Official Documentation — https://react.dev/
3.	FreeCodeCamp Full Stack JavaScript Tutorials — https://www.freecodecamp.org/
4.	Docker Documentation — https://docs.docker.com/
5.	MySQL Documentation — https://dev.mysql.com/doc/
6.	Postman Learning Center — https://learning.postman.com/
7.	Spring Initializr (https://start.spring.io/)

List of Experiments.

Sr No	List of Experiments	Hrs
01	Setup Java, Spring Boot, and React development environment.	01
02	Develop a static web page using HTML, CSS, and Bootstrap.	02
03	Build a dynamic front-end using React.js with API calls.	04
04	Develop a RESTful web service using Spring Boot.	02
05	Integrate MySQL database with Spring Boot application.	02
06	Implement JWT authentication in a Spring Boot application.	04
07	Connect React front-end with Spring Boot backend APIs.	04
08	Deploy the application using Docker and GitHub.	06

Sr No	List of Assignments	Hrs
01	Create a personal portfolio website using HTML, CSS, and React.	02
02	Develop a basic CRUD API using Spring Boot.	02
03	Implement a login and registration system with JWT.	02
04	Perform database operations using JPA Repository.	02
05	Build a single-page application (SPA) with React.js.	02
06	Deploy an integrated full-stack application on cloud/GitHub Pages.	02

Sr No	Capstone Mini- Project List (Sample Ideas)
01	**E-commerce Web Application** — Complete shopping system with admin dashboard.
02	**Online Library Management System** — For managing books, users, and borrowing transactions.
03	**Job Portal Application** — For job seekers and recruiters.
04	**Online Food Ordering System** — With user authentication, menu, and payment gateway.
05	**Student Course Registration System** — For colleges/universities.
06	**Event Management System** — Event creation, management, ticket booking.

Note:- Capstone Mini-Project Guidelines.

1. **Objective of the Capstone Mini-Project:-** The capstone Mini-Project aims to enable students to design, develop, deploy, and present a real-world full-stack application using the Java technology stack, applying concepts of frontend, backend, database management, RESTful APIs, containerization (Docker), security (JWT), and deployment practices.

   Students will work in a group of 4 students teams to build a functional application that solves a specific problem or fulfills a business need.

2. **Project Scope & Expectations**:- **Project Scale:** Should involve at least 2–3 core modules (e.g., User Management, Dashboard, Reporting). **Technology Stack: Frontend:** React.js (with API integration), **Backend:** Spring Boot REST API, **Database:** MySQL with JPA/Hibernate, **Security:** JWT-based authentication, **Deployment:** Docker containers, optional Docker Compose, **Integration:** End-to-end integration between frontend, backend, and database.**Testing:** API testing with Postman; frontend and backend error handling. **Documentation:** Source code, design documentation, API documentation, and user manual. **Presentation:** Live demo, project report, and presentation slides.

3. **Project Deliverables:- Project Proposal:** Problem statement, Objectives, Project scope, Technology stack.**Design Documentation:** System architecture diagram (MVC, API flow), Database schema (ER diagram), UI wireframes/mockups, **Implementation:** Frontend and backend code repositories (GitHub or equivalent), RESTful API endpoints, JWT authentication and role-based access control, Containerized deployment with Docker, **Testing:**Test cases and screenshots (Postman / frontend validation), **Final Report:** Introduction, methodology, results, testing summary, and conclusion, Challenges faced and solutions applied, Future scope. **Presentation:** 8–10 minutes demo, Architecture explanation, Live application demo, Q&A session.

Guidelines for Capstone Mini-Project

- Students shall form a group of 3 to 4 students, and a group shall not be allowed less than three or more than four students, as it is a group activity.
- Students should do the survey and identify needs, which shall be converted into a problem statement for the mini project in consultation with the faculty supervisor/head of department/internal committee of faculty.
- Students' hall submits an implementation plan in the form of a Gantt/PERT/CPM chart, which will cover the weekly activity of a mini-project.
- A log book to be prepared by each group, wherein the group can record weekly work progress, and the guide/supervisor can verify and record notes/comments.
- Faculty supervisor may give inputs to students during the mini project activity; however, focus shall be on self-learning.
- Students in a group shall understand the problem effectively, propose multiple solutions, and select the best possible solution in consultation with the guide/ supervisor.
- Students shall convert the best solution into a working model using various components of their domain areas and demonstrate.
- The solution to be validated with proper justification and a report to be compiled in a standard format of the University of Mumbai.

Guidelines for Assessment of Capstone Mini-Project:

Term Work

- The review/ progress monitoring committee shall be constituted by the head of departments of each institute. The progress of the mini project to be evaluated on continuous basis, minimum two reviews in each semester.
- In continuous assessment, focus shall also be on each individual student, assessment based on the individual's contribution in group activity, their understanding, and response to questions.
- Distribution of Term work marks for both semesters shall be as below;
  - Marks awarded by guide/supervisor based on above practical list and assignment completed by the students : 10
  - Marks awarded by guide/supervisor for Capstone Project Completion: :10
  - Quality of Project report & Project Review : 05

Review/progress monitoring committee may consider following points for assessment based of the semester project as mentioned in general guidelines.

- In this case in students' group shall complete project in all aspects including,
  - Identification of need/problem
  - Proposed final solution
  - Procurement of components/systems
  - Building prototype and testing
- Two reviews will be conducted for continuous assessment,
  - First shall be for the finalisation of problem and proposed solution
  - Second shall be for the implementation, testing and validation of solution.

Assessment criteria of Capstone Mini-Project.

Capstone Mini-Project shall be assessed based on following criteria;

1. Quality of survey/ need identification
2. Clarity of Problem definition based on need.
3. Innovativeness in solutions/ Novelty in solutions.
4. Feasibility of proposed problem solutions and selection of best solution
5. Collection of Dataset.
6. Cost effectiveness
7. Societal impact
8. Innovativeness

9. Cost effectiveness and Societal impact
10. Full functioning of working model as per stated requirements
11. Effective use of skill sets
12. Effective use of standard engineering norms
13. Contribution of an individual's as member or leader
14. Clarity in written and oral communication

Guidelines for Assessment of Capstone Mini-Project Practical/Oral Examination:

- Report should be prepared as per the guidelines issued by the University of Mumbai.
- Mini Project shall be assessed through a presentation and demonstration of working model by the student project group to a panel of Internal and External Examiners preferably from industry or research organisations having experience of more than five years approved by head of Institution.
- Students shall be motivated to publish a paper based on the work in Conferences/students competitions.

Assessment :
Term Work: Term Work shall consist of list of all practicals' based on the above list. Also, the Term work Journal must include at least 2 assignments and Mini-Project Report.

Term Work Marks: 50 Marks (Total marks) = 10 Marks (Experiment) + 5 Marks (Assignments) + 30 Marks (Capstone Mini- Project with full prototype/ product demo, testing, validation and Report) + 5 Marks (Attendance).

Oral Exam: An Oral exam will be held based on the Capstone Mini-Project.

Letter Grades and Grade Points:

Semester GPA/ Programme CGPA Semester/ Programme	% of Marks	Alpha-Sign/ Letter Grade Result	Grading Point
9.00 - 10.00	90.0 – 100	O (Outstanding)	10
8.00 - < 9.00	80.0 - < 90.0	A+ (Excellent)	9
7.00 - < 8.00	70.0 - < 80.0	A (Very Good)	8
6.00 - < 7.00	60.0 - < 70.0	B+ (Good)	7
5.50 - < 6.00	55.0 - < 60.0	B (Above Average)	6
5.00 - < 5.50	50.0 - < 55.0	C (Average)	5
4.00 - < 5.00	40.0 - < 50.0	P (Pass)	4
Below 4.00	Below 40.0	F (Fail)	0
Ab (Absent)	-	Ab (Absent)	0

Sd/-

Dr. Vaishali D. Khairnar
BoS-Chairman-Information Technology
Faculty of Technology

Sd/-

Dr. Deven Shah
Associate Dean
Faculty of Science & Technology

Sd/-

Prof. Shivram S. Garje
Dean
Faculty of Science & Technology