Program Structure for Second Year
Engineering Semester III & IV
UNIVERSITY OF MUMBAI
(With Effect from 2020-2021)

Semester III

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	**Pract.**	**Tut.**	**Theory**	**Pract.**	**Tut.**	**Total**
ITC301	Engineering Mathematics-III	3	--	1	3	--	1	4
ITC302	Data Structure and Analysis	3		--	3		--	3
ITC303	Database Management System	3	--	--	3	--	--	3
ITC304	Principle of Communication	3	--	--	3	--	--	3
ITC305	Paradigms and Computer Programming Fundamentals	3	--	--	3	--	--	3
ITL301	Data Structure Lab	--	2	--	--	1	--	1
ITL302	SQL Lab	--	2	--	--	1	--	1
ITL303	Computer programming Paradigms Lab	--	2	--	--	1	--	1
ITL304	Java Lab (SBL)	--	4	--	--	2	--	2
ITM301	Mini Project – 1 A for Front end /backend Application using JAVA	--	4[$]	--	--	2	--	2
Total		**15**	**14**	**1**	**15**	**07**	**1**	**23**

Course Code	Course Name	Examination Scheme							
		Theory					**Term Work**	**Pract/ oral**	**Total**
		Internal Assessment			**End Sem. Exam**	**Exam. Duration (in Hrs)**			
		Test 1	**Test2**	**Avg.**					
ITC301	Engineering Mathematics-III	20	20	20	80	3	25	--	125
ITC302	Data Structure and Analysis	20	20	20	80	3	--	--	100
ITC303	Database Management System	20	20	20	80	3	--	--	100
ITC304	Principle of Communication	20	20	20	80	3	--	--	100
ITC305	Paradigms and Computer Programming Fundamentals	20	20	20	80	3	--	--	100
ITL301	Data Structure Lab	--	--	--	--	--	25	25	50
ITL302	SQL Lab	--	--	--	--	--	25	25	50
ITL303	Computer programming Paradigms Lab	--	--	--	--	--	25	25	50
ITL304	Java Lab (SBL)	--	--	--	--	--	25	25	50
ITM301	Mini Project – 1 A for Front end /backend Application using JAVA	--	--	--	--	--	25	25	50
Total		**--**	**--**	**100**	**400**	**--**	**150**	**125**	**775**

$ indicates work load of Learner (Not Faculty), for Mini-Project. Students can form groups with minimum 2 (Two) and not more than 4 (Four) Faculty Load : 1 hour per week per four groups.

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	TW/Pract	Tut.	Total
ITC301	Engineering Mathematics-III	03	-	01	03	-	01	04

Course Code	Course Name	Examination Scheme							
		Theory				Term Work	Pract	Oral	Total
		Internal Assessment			End Sem Exam				
		Test1	Test2	Avg of Test 1 & 2					
ITC301	Engineering Mathematics-III	20	20	20	80	25	-	-	125

Pre-requisite: Engineering Mathematics-I, Engineering Mathematics-II

Course Objectives:

Sr. No.	Course Objectives
The course aims:	
1	To familiarize with the Laplace Transform, Inverse Laplace Transform of various functions, and its applications.
2	To acquaint with the concept of Fourier series, its complex form and enhance the 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 like 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

Module	**Detailed Contents**	**Hours**	**CO Mapping**
01	**Module: Laplace Transform** 1.1 Definition of Laplace transform, Condition of Existence of Laplace transform, 1.2 Laplace Transform (L) of Standard Functions like e^{at}, $sin(at)$, $cos(at)$, $sinh(at)$, $cosh(at)$ and $t^n, n \geq 0$. 1.3 Properties of Laplace Transform: Linearity, First Shifting Theorem, Second Shifting Theorem, change of scale Property, multiplication by *t*, Division by *t*, Laplace Transform of derivatives and integrals (Properties without proof). 1.4 Evaluation of real integrals by using Laplace Transformation. **Self-learning Topics:** Heaviside's Unit Step function, Laplace Transform. of Periodic functions, Dirac Delta Function.	7	CO1
02	**Module: Inverse Laplace Transform** 2.1 Inverse Laplace Transform, Linearity property, use of standard formulae to find inverse Laplace Transform, finding Inverse Laplace transform using derivatives, 2.2 Partial fractions method to find inverse Laplace transform. 2.3 Inverse Laplace transform using Convolution theorem (without proof) **Self-learning Topics:** Applications to solve initial and boundary value problems involving ordinary differential equations	6	CO1, CO2
03	**Module: Fourier Series:** 3.1 Dirichlet's conditions, Definition of Fourier series and Parseval's Identity(without proof) 3.2 Fourier series of periodic function with period 2π and $2l$, Fourier series of even and odd functions 3.4 Half range Sine and Cosine Series. **Self-learning Topics:** Complex form of Fourier Series, orthogonal and orthonormal set of functions, Fourier Transform.	7	CO3

04	**Module: Complex Variables:** 4.1 Function *f*(z) of complex variable, limit, continuity and differentiability of *f*(z) Analytic function, necessary and sufficient conditions for *f*(z) to be analytic (without proof), 4.2 Cauchy-Riemann equations in cartesian coordinates (without proof) 4.3 Milne-Thomson method to determine analytic function *f*(z) when real part (u) or Imaginary part (v) or its combination (u+v or u-v) is given. 4.4 Harmonic function, Harmonic conjugate and orthogonal trajectories **Self-learning Topics:** Conformal mapping, linear, bilinear mapping, cross ratio, fixed points and standard transformations	7	CO4
05	**Module: Statistical Techniques** 5.1 Karl Pearson's Coefficient of correlation (r) 5.2 Spearman's Rank correlation coefficient (R) (with repeated and non-repeated ranks) 5.3 Lines of regression 5.4 Fitting of first and second degree curves. **Self-learning Topics:** Covariance, fitting of exponential curve.	6	CO5
06	**Module: Probability** 6.1 Definition and basics of probability, conditional probability, 6.2 Total Probability Theorem and Baye's theorem 6.3 Discrete and continuous random variable with probability distribution and probability density function. 6.4 Expectation of random variables with mean, variance and standard deviation, moment generating function up to four moments. **Self-learning Topics:** Skewness and Kurtosis of distribution (data)	6	CO6

References:

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. Jain and S. R. K. Iyengar, Narosa publication,
4. Complex Variables and Applications, Brown and Churchill, McGraw-Hill education.
5. Probability, Statistics and Random Processes, T. Veerarajan, McGraw-Hill education.
6. Theory and Problems of Fourier Analysis with applications to BVP, Murray Spiegel, Schaum's Outline Series.

Online References:

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

Term Work:

General Instructions:

1. Students must be encouraged to write at least 6 class tutorials on 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 minutes. This should be considered as mini project in Engineering 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)	05 marks
2.	Class Tutorials on entire syllabus	10 marks
3.	Mini project	10 marks

Assessment:

Internal Assessment Test:
Assessment consists of two class tests of 20 marks each. The first class test (Internal Assessment I) is to be conducted when approx. 40% syllabus is completed and second class test (Internal Assessment II) when additional 35% syllabus is completed. Duration of each test shall be one hour.

End Semester Theory Examination:

1. Question paper will comprise of total 06 questions, each carrying 20 marks.
2. Total 04 questions need to be solved.
3. Question No: 01 will be compulsory and based on entire syllabus wherein 4 sub-questions of 5 marks each will be asked.
4. Remaining questions will be randomly selected from all the modules.
5. Weightage of each module will be proportional to number of respective lecture hours as mentioned in the syllabus.

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical /Oral	Tutorial	Total
ITC302	Data Structure and Analysis	03	--	--	03	--	--	03

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Pract. /Oral	Total
		Internal assessment			End Sem. Exam			
		Test1	Test 2	Avg.				
ITC302	Data Structure and Analysis	20	20	20	80	--	--	100

Course Objectives:

Sr. No.	Course Objectives
The course aims:	
1	The fundamental knowledge of data structures.
2	The programming knowledge which can be applied to sophisticated data structures.
3	The fundamental knowledge of stacks queue, linked list etc.
4	The fundamental knowledge of Trees, Graphs etc.
5	The fundamental knowledge of different sorting, searching, hashing and recursion techniques
6	The real time applications for stacks, queue, linked list, trees, graphs etc.

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	Classify and Apply the concepts of stacks, queues and linked list in real life problem solving.	L1, L2, L3
2	Classify, apply and analyze the concepts trees in real life problem solving.	L2, L3,L4
3	Illustrate and justify the concepts of graphs in real life problem solving.	L3, L5
4	List and examine the concepts of sorting, searching techniques in real life problem solving.	L2, L3, L4
5	Use and identify the concepts of recursion, hashing in real life problem solving.	L3, L4
6	Examine and justify different methods of stacks, queues, linked list, trees and graphs to various applications.	L3, L4, L5

Prerequisite: C Programming

DETAILED SYLLABUS:

Sr. No.	Module	Detailed Content	Hours	CO Mapping
0	Prerequisite	Defining, Declaring and Initialization of structure variables. Accessing members of a structure, Array of structures, Nested structures, Pointers to structures. Passing structure, structure members, structure arrays and pointer to structure as function parameters. Self-referential structures.	**02**	---
I	Introduction to Stacks, Queues and Linked Lists	Introduction to Data Structures: Linear and Non Linear Data Structures, Static and Dynamic Data Structures. Concept of Stack and Queue. Array Implementation of Stack and Queue, Circular Queue, Double Ended Queue, Priority Queue. Concept of Linked Lists. Singly linked lists, doubly linked lists and circular linked lists. Insertion, deletion, update and copying operations with Singly linked lists, doubly linked lists and circular linked lists. Reversing a singly linked list. **Self-learning Topics:** Linked List Implementation of Stack, Linked List implementation of Queue, Circular Queue, Double Ended Queue, Priority Queue.	**08**	CO1
II	Trees	Introduction to Trees: Terminology, Types of Binary trees. Non recursive Preorder, in-order and post-order traversal. Creation of binary trees from the traversal of binary trees. Binary search tree: Traversal, searching, insertion and deletion in binary search tree. Threaded Binary Tree: Finding in-order successor and predecessor of a node in threaded tree. Insertion and deletion in threaded binary tree. AVL Tree: Searching and traversing in AVL trees. Tree Rotations: Right Rotation, Left Rotation. Insertion and Deletion in an AVL Tree. B-tree: Searching, Insertion, Deletion from leaf node and non-leaf node. B+ Tree, Digital Search Tree, Game Tree & Decision Tree **Self-learning Topics:** Implementation of AVL and B+ Tree	**07**	CO1, CO 2
III	Graphs	Introduction to Graphs: Undirected Graph, Directed Graph, graph terminology, Connectivity in Undirected and Directed Graphs. Spanning tree. Representation of graph: adjacency matrix, adjacency list, Transitive closure of a directed graph and path matrix.	**05**	CO1, CO3

		Traversals: Breadth First Search, Depth First Search. **Self-learning Topics:** Implementation of BFS, DFS		
IV	Recursion and Storage Management	Recursion: Writing a recursive function, Flow of control in recursive functions, Winding and unwinding phase, Recursive data structures, Implementation of recursion. Tail recursion. Indirect and Direct Recursion. Storage Management: Sequential Fit Methods: First Fit, Best Fit and Worst Fit methods. Fragmentation, Freeing Memory, Boundary Tag Method. Buddy Systems: Binary Buddy System, Fibonacci Buddy System. Compaction, Garbage Collection. **Self-learning Topics:** Implementation of recursion function.	**06**	CO5
V	Searching and Sorting	Searching: Sequential Search, Binary Search. Hashing: Hash Functions: Truncation, Mid-square Method, Folding Method, Division Method. Collision Resolution: Open Addressing: Linear Probing, Quadratic Probing, Double Hashing, Separate Chaining Bucket Hashing. Analysis of all searching techniques Sorting: Insertion sort, Selection sort, Merge sort, Quick sort and Radix sort. Analysis of all sorting techniques **Self-learning Topics:** Implementation of different sorting techniques and searching.	**05**	CO 4, CO5
VI	Applications of Data Structures	Applications of Linked Lists: Addition of 2 Polynomials and Multiplication of 2 polynomials. Applications of Stacks: Reversal of a String, Checking validity of an expression containing nested parenthesis, Function calls, Polish Notation: Introduction to infix, prefix and postfix expressions and their evaluation and conversions. Application of Queues: Scheduling, Round Robin Scheduling Applications of Trees: Huffman Tree and Heap Sort. Applications of Graphs: Dijkstra's Algorithm, Minimum Spanning Tree: Prim's Algorithm, Kruskal's Algorithm. **Self-learning Topics:** Implementation of applications for Stack, Queues, Linked List, Trees and Graph.	**06**	CO6

Text Books:

1. S. K Srivastava, Deepali Srivastava; Data Structures through C in Depth; BPB Publications; 2011.
2. Yedidya Langsam, Moshej Augenstein, Aaron M. Tenenbaum; Data Structure Using C & C++; Prentice Hall of India; 1996.
3. Reema Thareja; Data Structures using C; Oxford.

References:

1. Ellis Horowitz, Sartaj Sahni; Fundamentals of Data Structures; Galgotia Publications; 2010.
2. Jean Paul Tremblay, Paul G. Sorenson; An introduction to data structures with applications; Tata McGrawHill; 1984.
3. Rajesh K. Shukla; Data Structures using C and C++; Wiley India; 2009.

Online References:

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

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 20 marksQ.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** need to be answered

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical /Oral	Tutorial	Total
ITC303	Database Management System	03	--	--	03	--	--	03

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Pract. /Oral	Total
		Internal assessment			End Sem. Exam			
		Test1	Test 2	Avg.				
ITC303	Database Management System	20	20	20	80	--	--	100

Course Objectives:

Sr. No.	Course Objectives
The course aims:	
1	To learn the basics and understand the need of database management system.
2	To construct conceptual data model for real world applications
3	To Build 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:

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	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

Prerequisite: C Programming

DETAILED SYLLABUS:

Sr. No.	Module	Detailed Content	Hours	CO Mapping
0	Prerequisite	**Comment**Basic knowledge of operating systems and file systems, Any programming	**02**	--
I	**Database System Concepts and 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 **Self-learning Topics:** Identify the types of Databases.	**05**	CO1
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 upto 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 situation which may occur for a database designed in module V.	**07**	CO6

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 (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 20 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** need to be answered

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical /Oral	Tutorial	Total
ITC304	Principle of Communication	03	--	--	03	--	--	03

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Pract. /Oral	Total
		Internal assessment			End Sem. Exam			
		Test1	Test 2	Avg.				
ITC304	Principle of Communication	20	20	20	80	--	--	100

Course Objectives:

Sr. No.	Course Objectives
The course aims:	
1	Study the basic of Analog and Digital Communication Systems.
2	Describe the concept of Noise and Fourier Transform for analyzing communication systems.
3	Acquire the knowledge of different modulation techniques such as AM, FM and study the block diagram of transmitter and receiver.
4	Study the Sampling theorem and Pulse Analog and digital modulation techniques
5	Learn the concept of multiplexing and digital band pass modulation techniques
6	Gain the core idea of electromagnetic radiation and propagation of waves.

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	Describe analog and digital communication systems	L1,L2
2	Differentiate types of noise, analyses the Fourier transform of time and frequency domain.	L1, L2, L3, L4
3	Design transmitter and receiver of AM, DSB, SSB and FM.	L1,L2,L3,L4
4	Describe Sampling theorem and pulse modulation systems.	L1,L2,L3
5	Explain multiplexing and digital band pass modulation techniques.	L1, L2
6	Describe electromagnetic radiation and propagation of waves.	L1,L2

Prerequisite: Basic of electrical engineering

DETAILED SYLLABUS:

Sr. No.	Module	Detailed Content	Hours	CO Mapping
0	Prerequisite	Terminologies in communication systems, analog and digital electronics	**02**	
I	**Introduction**	Basics of analog communication and digital communication systems (Block diagram), Electromagnetic Spectrum and application, Types of Communication channels. **Self-learning Topics:** Applications areas of analog and digital communication.	**03**	**CO1**
II	**Noise and Fourier Representation of Signal and System**	Basics of signal representation and analyses, Introduction to Fourier Transform, its properties (time and frequency shifting, Fourier transform of unit step, delta and gate function. Types of Noise, Noise parameters –Signal to noise ratio, Noise factor, Noise figure, Friss formula and Equivalent noise temperature. **Self-learning Topics:** Practice Numerical on above topic.	**06**	**CO2**
III	**Amplitude and Angle modulation Techniques.**	Need for modulation, Amplitude Modulation Techniques: DSBFC AM,DSBSC-AM, SSB SC AM- block diagram spectrum, waveforms, bandwidth, Power calculations. Generation of AM using Diode, generation of DSB using Balanced modulator, Generation of SSB using Phase Shift Method. AM Transmitter (Block Diagram) AM Receivers – Block diagram of TRF receivers and Super heterodyne receiver and its characteristics- Sensitivity, Selectivity, Fidelity, Image frequency and its rejection and double spotting **Angle Modulation** **FM:** Principle of FM- waveforms, spectrum, bandwidth. Pre- emphasis and de-emphasis in FM, FM generation: Direct method –Varactor diode Modulator, Indirect method (Armstrong method) block diagram and waveforms. FM demodulator: Foster Seeley discriminator, Ratio detector. **Self-learning Topics:** Use of AM and FM in Modern Communication Technology. Challenges faced by radio business.	**12**	**CO1, CO2, CO3**
IV	**Pulse Analog Modulation and Digital Modulation**	Sampling theorem for low pass and band pass signals with proof, Anti- aliasing filter, PAM, PWM and PPM generation and Degeneration. Quantization process, Pulse code modulation, Delta modulation, Adaptive delta modulation. Introduction to Line Codes and ISI.	**08**	**CO1, CO2, CO4**

		Self-learning Topics: Implementation of Pulse code modulation and demodulation.		
V	**Multiplexing and Digital Band Pass Modulation Techniques**	Principle of Time Division Multiplexing, Frequency Division Multiplexing , Orthogonal Frequency Division Multiplexing and its applications .ASK, FSK, PSK QPSK Generation and detection. **Self-learning Topics:** Implement TDM, FDM, OFDM.	**04**	**CO1, CO2, CO5**
VI	**Radiation and Propagation of Waves**	Electromagnetic radiation, fundamentals, types of propagation, ground wave, sky wave, space wave tropospheric scatter propagation **Self-learning Topics:** List the real time examples for different types of propagation waves.	**04**	**CO6**

Text Books:

[1]. George Kennedy, Bernard Davis, SRM Prasanna, Electronic Communication Systems, Tata McGraw Hill, 5th Ed
[2]. Simon Haykin, Michael Moher, Introduction to Analog & Digital Communications, Wiley India Pvt. Ltd., 2nd Ed.
[3].Wireless Communication and Networking, Vijay Garg

References:

[1]. Wayne Tomasi, Electronic Communications Systems, Pearson Publication, 5th Ed.
[2]. B P Lathi, Zhi Ding, Modern Digital and Analog Communication Systems, Oxford University
[3]. Herbert Taub, Donald L Schilling, Goutam Saha, Principles of Communication Systems, Tata McGraw Hill, 3rdEd.
[4]. K Sam Shanmugam, Digital and Analog Communication Systems, Wiley India Pvt. Ltd, 1st Ed.

Online References:

Sr. No.	Website Name
1.	https://www.nptel.ac.in
2.	https://www.classcentral.com
3.	http://www.vlab.co.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 20 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** need to be answered

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical /Oral	Tutorial	Total
ITC305	Paradigms and Computer Programming Fundamentals	03	--	--	03	--	--	03

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Pract. /Oral	Total
		Internal assessment			End Sem. Exam			
		Test1	Test 2	Avg.				
ITC305	Paradigms and Computer Programming Fundamentals	20	20	20	80	--	--	100

Course Objectives:

Sr. No.	Course Objectives
The course aims:	
1	To introduce various programming paradigms and the basic constructs that underline any programming language.
2	To understand data abstraction and object orientation
3	To introduce the basic concepts of declarative programming paradigms through functional and logic programming.
4	To design solutions using declarative programming paradigms through functional and logic programming.
5	To introduce the concepts of concurrent program execution.
6	To understand use of scripting language for different problem domains

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	Understand and Compare different programming paradigms.	L1, L2
2	Understand the Object Oriented Constructs and use them in program design.	L1, L2
3	Understand the concepts of declarative programming paradigms through functional and logic programming.	L1, L2
4	Design and Develop programs based on declarative programming paradigm using functional and/or logic programming.	L5, L6
5	Understand the role of concurrency in parallel and distributed programming.	L1, L2
6	Understand different application domains for use of scripting languages.	L1. L2

Prerequisite: Students must have learned C Programming (FEC205 and FEL204),

DETAILED SYLLABUS:

Sr. No.	Module	Detailed Content	Hours	CO Mapping
0	Prerequisite	Compilation and interpretation Focus on overview of compilation steps.	**02**	CO1
I	Introduction to Programming Paradigms and Core Language Design Issues	Introduction to different programming paradigms. Names, Scopes, and Bindings, Scope Rules, Storage Management. Type Systems, Type Checking, Equality Testing and Assignment. Subroutine and Control Abstraction: Stack Layout, Calling sequence, parameter passing Generic subroutines and modules. Exception handling, Coroutines and Events. **Self-Learning Topic:** Implementation of basic concepts using programming language.	**10**	CO1
II	Imperative Paradigm: Data Abstraction in Object Orientation	Grouping of data and Operations- Encapsulation, Overloading, Polymorphism, Inheritance, Initialization and Finalization, Dynamic Binding. **Self-Learning Topic:** Implementation of OOP concepts using preferrably C++ and Java language.	**05**	CO2
III	Declarative Programming Paradigm: Functional Programming	Introduction to Lambda Calculus, Functional Programming Concepts, Evaluation order, Higher order functions, I/O-Streams and Monads. **Self-Learning Topic:** Implementation of programs using functional programming Language Haskel can refer to hacker rank website for problem statements.	**07**	CO3, CO4
IV	Declarative Programming Paradigm: Logic Programming	Logic Programming with PROLOG - Resolution and Unification, Lists, Arithmetic execution order, imperative control flow, database manipulation, PROLOG facilities and deficiencies. **Self-Learning Topic:** Identification of different application domains for use of Prolog and Logic programming	**06**	CO3, CO4
V	Alternative Paradigms: Concurrency	Concurrent Programming Fundamentals, Implementing synchronisation, Message Passing - Background and Motivation, Multithreaded programs, Communication and Synchronization, Language and Libraries, Thread creation Syntax. **Self-Learning Topic:** Study Implementation of concurrency concepts for real time application.	**04**	CO5
VI	Alternative Paradigms: Scripting Languages	Common characteristics, Different Problem domains for using scripting,Use of scripting in Web development–server and clients side scripting, Innovative features of scripting languages - Names and Scopes, string and pattern manipulation, data types, object orientation.	**05**	CO6

		Self-Learning Topic: Review small client server application code in any scripting language to realise applicability of features learned in Module.		

Text Books:

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. Programming Languages: Concepts and Constructs; 2nd Edition, Ravi Sethi, Pearson Education Asia, 1996.

References:

1. Harold Abelson and Gerald Jay Sussman with Julie Sussman foreword by Alan J. Perlis, Structure and Interpretation of Computer Programs (2nd Edition) (February 2, 2016)
2. Programming Languages: Design and Implementation (4th Edition), by Terrence W. Pratt, Marvin V. Zelkowitz, Pearson, 2000
3. Rajkumar Buyya, Object-oriented Programming with Java: Essentials and Applications, Tata McGraw Hill Education Private Limited
4. Max Bramer, Logic Programming with Prolog, Springer ISBN-13: 978-1852-33938-8

Online References:

Sr No	Website Name	Link
1	Principles of programming Languages (Videos)	https://nptel.ac.in/courses/106/102/106102067/
2	Edx course Paradigms of Computer Programming – Fundamentals	https://www.classcentral.com/course/edx-paradigms-of-computer-programming-fundamentals-2298
3	Udemy Couses	https://www.udemy.com

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 20 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 need to be answered

Lab Code	Lab Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
ITL301	Data Structure Lab	--	02	--	--	01	--	01

Lab Code	Lab Name	Examination Scheme						
		Theory Marks				Term Work	Pract. /Oral	Total
		Internal assessment			End Sem. Exam			
		Test1	Test 2	Avg.				
ITL301	Data Structure Lab	--	--	--	--	25	25	50

Lab Objectives:

Sr. No.	Lab Objectives
The Lab experiments aims:	
1	To use data structures as the introductory foundation for computer automation to engineering problems.
2	To use the basic principles of programming as applied to complex data structures.
3	To learn the principles of stack, queue, linked lists and its various operations.
4	To learn fundamentals of binary search tree, implementation and use of advanced tree like AVL, B trees and graphs.
5	To learn about searching, hashing and sorting.
6	To learn the applications of linked lists, stacks, queues, trees and graphs.

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 and use the basic concepts and principles of various linked lists, stacks and queues.	L1, L2, L3
2	Understand the concepts and apply the methods in basic trees.	L1, L2
3	Use and identify the methods in advanced trees.	L3, L4
4	Understand the concepts and apply the methods in graphs.	L2, L3
5	Understand the concepts and apply the techniques of searching, hashing and sorting	L2, L3
6	Illustrate and examine the methods of linked lists, stacks, queues, trees and graphs to various real time problems	L3, L4

Prerequisite: C Programming

Hardware & Software Requirements:

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

DETAILED SYLLABUS:

Sr. No.	**Module**	**Detailed Content**	**Hours**	**LO Mapping**
0	Prerequisite	Introduction of C programming language.	**02**	----
I	Stacks, Queues and Linked Lists	Array Implementation of Stack and Queue. Insertion, deletion operations with Singly linked lists Insertion, deletion operations Doubly linked lists Insertion, deletion operations Circular linked lists. Reversing a singly linked list. *** Linked List implementation of Stack and Queue**	**04**	LO 1
II	Trees	*** Implementation of operations (insertion, deletion, counting of nodes, counting of leaf nodes etc.) in a binary search tree.** Implementation of insertion, deletion and traversal for fully in-threaded binary search tree.	**04**	LO 2
III	Advanced Trees	*** Implementation of AVL tree.** Implementation of operations in a B tree.	**04**	LO 3
IV	Graphs	Implementation of adjacency matrix creation. Implementation of addition and deletion of edges in a directed graph using adjacency matrix. Implementation of insertion and deletion of vertices and edges in a directed graph using adjacency list.	**04**	LO 4
V	Searching and Sorting	Implementation of Heap Sort Implementation of Binary Search. Implementation of Selection sort, Bubble sort, Insertion sort, Quick sort	**04**	LO 5

VI	Applications of Data Structures	*** Implementation of infix to postfix conversion and evaluation of postfix expression** *** Implementation of Josephus Problem using circular linked list** *** Implementation of traversal of a directed graph through BFS and DFS.** Implementation of finding shortest distances using Dijkstra's algorithm ***Implementation of hashing functions with different collision resolution techniques**	**04**	LO 6

Text Books:

1. S. K Srivastava, Deepali Srivastava; Data Structures through C in Depth; BPB Publications; 2011.
2. Yedidya Langsam, Moshej Augenstein, Aaron M. Tenenbaum; Data Structure Using C & C++; Prentice Hall of India; 1996.
3. Reema Thareja; Data Structures using C; Oxford.

References:

1. Ellis Horowitz, Sartaj Sahni; Fundamentals of Data Structures; Galgotia Publications; 2010.
2. Jean Paul Tremblay, Paul G. Sorenson; An introduction to data structures with applications; Tata McGrawHill; 1984.
3. Rajesh K. Shukla; Data Structures using C and C++; Wiley India; 2009.

Term Work: Term Work shall consist of at least 10 to 12 practical's based on the 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 the above syllabus.

Lab Code	Lab Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
ITL302	SQL Lab	--	02	--	--	01	--	01

Lab Code	Lab Name	Examination Scheme						
		Theory Marks				Term Work	Pract. /Oral	Total
		Internal assessment			End Sem. Exam			
		Test1	Test 2	Avg.				
ITL302	SQL Lab	--	--	--	--	25	25	50

Lab Objectives:

Sr. No.	Lab Objectives
The Lab experiments aims:	
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:

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	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 Programming

Hardware & Software Requirements:

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

DETAILED SYLLABUS:

Sr. No.	Detailed Content	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.	**02**	LO3, LO4
12.	Implementation and demonstration of Transaction and Concurrency control techniques using locks.	**02**	LO6

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

Term Work:

Term Work shall consist of at least 10 Practical's based on the above list, but not limited to. Also, Term work Journal must include at least 2 assignments:

The first assignment may be based on: Relational Algebra and Second may be based on Transactions

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 the above syllabus.

Lab Code	Lab Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
ITL303	Computer programming Paradigms Lab	--	02	--	--	01	--	01

Lab Code	Lab Name	Examination Scheme						
		Theory Marks				Term Work	Pract. /Oral	Total
		Internal assessment			End Sem. Exam			
		Test1	Test 2	Avg.				
ITL303	Computer programming Paradigms Lab	--	--	--	--	25	25	50

Lab Objectives:

Sr. No.	Lab Objectives
The Lab experiments aims:	
1	Understand data abstraction and object orientation
2	Design and implement declarative programs in functional and logic programming languages
3	Introduce the concepts of concurrent program execution
4	Understand run time program management
5	Understand how to implement a programming solution using different programming paradigms.
6	Learn to compare implementation in different programming paradigms.

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	Implement Object Oriented concepts in C++.	L1, L2, L3
2	Design and Develop solution based on declarative programming paradigm using functional and logic programming.	L6
3	Understand the multi threaded programs in Java and C++	L1, L2
4	Understand the need and use of exception handling and garbage collection in C++ and JAVA	L2, L3
5	Implement a solution to the same problem using multiple paradigms.	L6
6	Compare the implementations in multiple paradigms at coding and	L4

	execution level.	

Prerequisite: Students must have learned C Programming (FEC205 and FEL204)

Hardware & Software Requirements:

Hardware Requirement:	**Software requirement:**
PC i3 processor and above	C++ compiler, Java Languge support, SWI Prolog, GHC Compiler.

DETAILED SYLLABUS:

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	Imperative Paradigm: Data Abstraction in Object Orientation	At least two Programming Implementations Preferably in C++ to demonstrate concepts like - Encapsulation, Inheritance, Initialization and Finalization, Dynamic Binding.	**05**	LO1
II	Declarative Programming Paradigm: Functional Programming	Tutorial Introduction to Haskell programming environment Tutorial exercise on operators, types etc. in Haskell At least 5 Haskell Programs to demonstrate Functional Programming Concepts. Sample Programs but not limited to: ◦ Implement 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. Hint: the library function null :: [a]-> Bool can be used to test if a list is empty. ◦ Simple List Comprehension ◦ Higher-Order Functions ◦ Write recursive function to multiply two natural numbers that uses pre defined add funion. ◦ Implement the game of nim in Haskell to apply list processing. ◦ Haskell code to represent infinite list e.g. fibobacci series ◦ Implement simple Calculator Students should clearly understand the syntax and the execution of the Functional Implementation using Haskell.	**06**	LO2

III	Declarative Programming Paradigm: Logic Programming	Tutorial Installation and working of SWI Prolog Environment Implement at least 5 Prolog programs to understand declarative programming concepts. Students should clearly understand the syntax and the execution of the Prolog code Implementation.	**05**	LO2
IV	Alternative Paradigms: Concurrency	At least two Programs preferably in c++ and java to demonstrate Thread management and synchronization	**02**	LO4
V	Run Time Program Management	A Program to understand Exception handling and Garbage collection, preferably in C++ and JAVA Students should understand the syntactic differences in the solutions in both Object Oriented Languages.	**02**	LO4
VI	Programming Assignment For comparative study of Different Paradigms	At Least two implementations each implemented on multiple paradigms like procedural, object oriented, functional, logic. The implementations should be done in a group of two/three students with appropriate difficulty level. Student should prepare small report and present the solution code and demonstrate execution for alternative solutions they build.	**04**	LO5, LO6

Text Books:

1. Scott M L, Programming Language Pragmatics, 3rd Edn., Morgan Kaufmann Publishers, 2009
2. Harold Abelson and Gerald Jay Sussman with Julie Sussman foreword by Alan J. Perlis, Structure and Interpretation of Computer Programs (2nd Edition)
3. Graham Hutton, Programming in Haskell, 2nd Edition, Cambridge University Press, 2016
4.

References:

1. Sethi R, Programming Languages Concepts and Constructs , 2nd Ed, Pearson Education
2. Yogesh Sajanikar, Haskell Cookbook, Packt Publishing, 2017

Online References:

Sr No	Website Description	Link
1	University Stuttgart Germany Lab Course on Programming Paradigms	http://software-lab.org/teaching/winter2019/pp/
2	Course at MIT Structure and Interpretation of Computer Programs [2019]	https://web.mit.edu/u/6.037
3	Edx Course Paradigms of Computer Programming – Fundamentals,	https://www.edx.org/course/paradigms-of-computer-programming-fundamentals
4	Tutorials point link for Haskel	https://www.tutorialspoint.com/haskell

Term Work: Term Work shall consist of at least 15 Practicals based on the above modules, but not limited to. Also, Term work Journal must include at least 3 tutorial reports and 01 report of programming assignment

as mentioned in module VI.

Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiments/Tutorials) + 5 Marks (Assignment write up) + 5 Marks (Attendance)

Practical & Oral Exam: An Oral & 1 Hr Practical exam will be held based on the above syllabus

Lab Code	Lab Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
ITL304	Java Lab (SBL)	--	04	--	--	02	--	02

Lab Code	Lab Name	Examination Scheme						
		Theory Marks				Term Work	Pract. /Oral	Total
		Internal assessment			End Sem. Exam			
		Test1	Test 2	Avg.				
ITL304	Java Lab (SBL)	--	--	--	--	25	25	50

Lab Objectives:

Sr. No.	Lab Objectives
The Lab experiments aims:	
1	To understand the concepts of object-oriented paradigm in the Java programming language.
2	To understand the importance of Classes & objects along with constructors, Arrays ,Strings and vectors
3	To learn the principles ofinheritance, interface and packages and demonstrate the concept of reusability for faster development.
4	To recognize usage of Exception Handling, Multithreading, Input Output streams in various applications
5	To learn designing, implementing, testing, and debugging graphical user interfaces in Java using Swings and AWT components that can react to different user events.
6	To develop graphical user interfaces using JavaFX controls.

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	Explain the fundamental concepts of Java Programing.	L1, L2
2	Use the concepts of classes, objects, members of a class and the relationships among them needed for a finding the solution to specific problem.	L3
3	Demonstrate how to extend java classes and achieve reusability using Inheritance, Interface and Packages.	L3
4	Construct robust and faster programmed solutions to problems using concept of Multithreading, exceptions and file handling	L3
5	Design and develop Graphical User Interface using Abstract Window Toolkit and Swings along with response to the events.	L6
6	Develop Graphical User Interface by exploring JavaFX framework based on MVC architecture.	L6

Prerequisite: Basics of Computer Programming

Hardware & Software Requirements:

Hardware Requirements	Software Requirements	Other Requirements
PC With Following Configuration 1. Intel PIV Processor 2. 2 GB RAM 3. 500 GB Harddisk 4. Network interface card	1. Windows or Linux Desktop OS 2. JDK 1.8 or higher 3. Notepad ++ 4.JAVA IDEs like Netbeans or Eclipse	1. Internet Connection for installing additional packages if required

DETAILED SYLLABUS:

Sr. No.	Module	Detailed Content	Hours	LO Mapping
0	Prerequisite	Basics of Computer Programming.	**02**	-
I	Java Fundamentals	**Overview of procedure and object oriented Programming**, Java Designing Goals and Features of Java Language. **Introduction to the principles of object-oriented programming**: Classes, Objects, Abstraction, Encapsulation, Inheritance, Polymorphism. Keywords, Data types, Variables, Operators, Expressions, Types of variables and methods. **Control Statements**: If Statement, If-else, Nested if, switch Statement, break, continue. **Iteration Statements:** for loop, while loop, and do-while loop (Perform any 2 programs that covers Classes, Methods, Control structures and Looping statements) 1) Implement a java program to calculate gross salary & net salary taking the following data. Input: empno, empname, basic Process: DA=70% of basic HRA=30% of basic CCA=Rs240/- PF=10% of basic PT= Rs100/- 2) Five Bikers Compete in a race such that they drive at a constant speed which may or may not be the same as the other. To qualify the race, the speed of a racer must be more than the average speed of all 5 racers. Write a Java program to take as input the speed of each racer and print back the speed of qualifying racers. 3) Write a Java program that prints all real solutions to the quadratic equation $ax^2+bx+c = 0$. Read in a, b, c and use the quadratic formula. If the discriminate b^2-4ac is negative, display a message stating that there are no real solutions? 4) Write a Menu driven program in java to implement simple banking application. Application should read	**07**	LO1

		the customer name, account number, initial balance, rate of interest, contact number and address field etc. Application should have following methods. 1. createAccount() 2. deposit() 3. withdraw() 4. computeInterest() 5. displayBalance() 5)Write a menu driven Java program which will read a number and should implement the following methods 1. factorial() 2. testArmstrong() 3. testPalindrome() 4. testPrime() 5. fibonacciSeries() 6) Create a Java based application to perform various ways of Method overloading.		
II	Classes, objects, Arrays and Strings	**Classes & Objects**: Reference Variables, Passing parameters to Methods and Returning parameters from the methods, Static members, Non-Static members Nested and Inner Classes. Static Initialization Block(SIB), Instance Initialization Block(IIB) **Constructors**: Parameterized Constructors, chaining of constructor, finalize() Method, Method overloading, Constructors Overloading. Recursion, Command-Line Arguments. Wrapper classes, InputBufferReader, OutputBufferReader, String Buffer classes, String functions. **Arrays & Vectors**: One and Two Dimensional arrays, Irregular arrays, dynamic arrays, Array List and Array of Object. (Perform any 3 programs that covers Classes & objects, Constructors, Command Line Arguments, Arrays/Vectors,String function and recursions). **Experiments:** 1) Write a program that would print the information (name, year of joining, salary, address) of three employees by creating a class named 'Employee'. The output should be as follows: Name Year of joining Address Robert 1994 64C- WallsStreat Sam 2000 68D- WallsStreat John 1999 26B- WallsStreat 2) Write a program to print the area of a rectangle by creating a class named 'Area' having two methods. First method named as 'setDim' takes length and breadth of rectangle as parameters and the second method named as 'getArea' returns the area of the rectangle. Length and breadth of rectangle are entered through keyboard. 3) Write a Java program to illustrate Constructor Chaining.	**07**	LO1 LO2

		4) Create a class 'Student' with three data members which are name, age and address. The constructor of the class assigns default values name as "unknown", age as '0' and address as "not available". It has two members with the same name 'setInfo'. First method has two parameters for name and age and assigns the same whereas the second method takes has three parameters which are assigned to name, age and address respectively. Print the name, age and address of 10 students. Hint - Use array of objects. 5) Write a java programs to add n strings in a vector array. Input new string and check whether it is present in the vector. If it is present delete it otherwise add it to the vector. 6) Print the sum, difference and product of two complex numbers by creating a class named 'Complex' with separate methods for each operation whose real and imaginary parts are entered by user. 7)Write menu driven program to implement recursive Functions for following tasks. a) To find GCD and LCM b) To print n Fibonacci numbers c) To find reverse of number d) To solve 1 +2+3+4+.....+(n- l)+n 8) Print Reverse Array list in java by writing our own function.		
III	Inheritance, Packages and Interfaces.	**Inheritance**: Inheritance Basics, Types of Inheritance in Java, member access, using Super- to call superclass Constructor, to access member of super class(variables and methods), creating multilevel hierarchy, Constructors in inheritance, method overriding, Abstract classes and methods, using final, Dynamic Method Dispatch **Packages**: Defining packages, creating packages and Importing and accessing packages **Interfaces**: Defining, implementing and extending interfaces, variables in interfaces, Default Method in Interface ,Static Method in interface, Abstract Classes vs Interfaces. (Perform any 3 programs covering Inheritance, Interfaces and Packages). **Experiments** 1) Create a Teacher class and derive Professor/ Associate_Professor/Assistant_Professor class from Teacher class. Define appropriate constructor for all the classes. Also define a method to display information of Teacher. Make necessary assumptions as required. 2) Create a class Book and define a display method to display book information. Inherit Reference_Book and Magazine classes from Book class and override display method of Book class in Reference_Book and Magazine classes. Make necessary assumptions required.	**10**	LO1 LO3

3) A university has two types of students — graduate students and research students. The University maintains the record of name, age and programme of every student. For graduate students, additional information like percentage of marks and stream, like science, commerce, etc. is recorded; whereas for research students, additionally, specialization and years of working experience, if any, is recorded. Each class has a constructor. The constructor of subclasses makes a call to constructor of the superclass. Assume that every constructor has the same number of parameters as the number of instance variables. In addition, every subclass has a method that may update the instance variable values of that subclass. All the classes have a function display_student_info( ), the subclasses must override this method of the base class. Every student is either a graduate student or a research student.
Perform the following tasks for the description given above using Java :

(i) Create the three classes with proper instance variables and methods, with suitable inheritance.
(ii) Create at least one parameterised constructor for each class.
(iii) Implement the display_student_info( ) method in each class.

4) An employee works in a particular department of an organization. Every employee has an employee number, name and draws a particular salary. Every department has a name and a head of department. The head of department is an employee. Every year a new head of department takes over. Also, every year an employee is given an annual salary enhancement. Identify and design the classes for the above description with suitable instance variables and methods. The classes should be such that they implement information hiding. You must give logic in support of your design. Also create two objects of each class.

5) Consider a hierarchy, where a sportsperson can either be an athlete or a hockey player. Every sportsperson has a unique name. An athlete is characterized by the event in which he/she participates; whereas a hockey player is characterised by the number of goals scored by him/her.
Perform the following tasks using Java :
(i)Create the class hierarchy with suitable instance variables and methods.
(ii) Create a suitable constructor for each class.
(iii) Create a method named display_all_info with suitable parameters. This method should display all the information about the object of a class.
(iv) Write the main method that demonstrates polymorphism.

6) Create an interface vehicle and classes like bicycle,

		car, bike etc, having common functionalities and put all the common functionalities in the interface. Classes like Bicycle, Bike, car etc implement all these functionalities in their own class in their own way 7) Create a class "Amount In Words" within a user defined package to convert the amount into words. (Consider amount not to be more than 100000).		
IV	Exception Handling, Multithreading, Input Output streams	**Exception Handling:** Exception-Handling Fundamentals, Exception Types, Exception class Hierarchy, Using try and catch, Multiple catch Clauses, Nested try Statements, throw, throws, finally , Java's Built-in Exceptions, Creating Your Own Exception Subclasses **Multithreaded Programming:** The Java Thread Model and Thread Life Cycle, Thread Priorities, Creating a Thread, Implementing Runnable, Extending Thread, Creating Multiple Threads, Synchronization: Using Synchronized Methods, The synchronized Statement **I/O Streams:** Streams, Byte Streams and Character, The Predefined Streams, Reading Console Input, Reading Characters, Reading Strings, Writing Console Output, Reading and Writing Files. (Perform any 3 programs that cover Exception Handling, Multithreading and I/O Streams). **Experiments:** 1) Write java program where user will enter loginid and password as input. The password should be 8 digit containing one digit and one special symbol. If user enter valid password satisfying above criteria then show "Login Successful Message". If user enter invalid Password then create InvalidPasswordException stating Please enter valid password of length 8 containing one digit and one Special Symbol. 2) Java Program to Create Account with 1000 Rs Minimum Balance, Deposit Amount, Withdraw Amount and Also Throws LessBalanceException. It has a Class Called LessBalanceException Which returns the Statement that Says WithDraw Amount(_Rs) is Not Valid. It has a Class Which Creates 2 Accounts, Both Account Deposite Money and One Account Tries to WithDraw more Money Which Generates a LessBalanceException Take Appropriate Action for the Same. 3) Create two threads such that one thread will print even number and another will print odd number in an ordered fashion. 4) Assume that two brothers, Joe and John, share a common bank account. They both can, independently, read the balance, make a deposit, and withdraw some	**10**	LO1 LO3 LO4

		money. Implement java application demonstrate how the transaction in a bank can be carried out concurrently. 5) You have been given the list of the names of the files in a directory. You have to select Java files from them. A file is a Java file if it's name ends with ".java". For e.g. File- "Names.java" is a Java file, "FileNames.java.pdf" is not. **Input:** test.java, ABC.doc, Demo.pdf, add.java, factorial.java, sum.txt **Output:** tset.java, add.java, factorial.java		
V	GUI programming- I (AWT, Event Handling, Swing)	**Designing Graphical User Interfaces in Java**: Components and Containers, Basics of Components, Using Containers, Layout Managers, AWT Components, Adding a Menu to Window, Extending GUI Features **Event-Driven Programming in Java**: Event-Handling Process, Event-Handling Mechanism, Delegation Modelof Event Handling, Event Classes, Event Sources, Event Listeners, Adapter Classes as Helper Classes in Event Handling. **Introducing Swing:** AWT vs Swings, Components and Containers, Swing Packages, A Simple Swing Application, Painting in Swing, Designing Swing GUI Application using Buttons, JLabels, Checkboxes, Radio Buttons, JScrollPane, JList, JComboBox, Trees, TablesScroll pane Menus and Toolbar (Perform any 3 programs that contain AWT, Event handling and Swing to build GUI application). 1)Write a Java program to implement Swing components namely Buttons, ,JLabels, Checkboxes, Radio Buttons, JScrollPane, JList, JComboBox, Trees, Tables Scroll pane Menus and Toolbars to design interactive GUI. 2) Write a program to create a window with four text fields for the name, street, city and pincode with suitable labels. Also windows contains a button MyInfo. When the user types the name, his street, city and pincode and then clicks the button, the types details must appear in Arial Font with Size 32, Italics. 3) Write a Java program to create a simple calculator using java AWT elements. .Use a grid layout to arrange buttons for the digits and basic operation +, -, /, *. Add a text felid to display the results. 4) Write a Java Program to create a Student Profile form using AWT controls. 5) Write a Java Program to simulate traffic signal light using AWT and Swing Components.	**12**	LO1 LO4 LO5

		6) Write a Java Program to create a color palette. Declare a grid of Buttons to set the color names. Change the background color by clicking on the color button. 7) Build a GUI program that allows the user to add objects to a collection and perform search and sort on that collection.(Hint. Use Swing components like JButton, JList, JFrame, JPanel and JOptionPane.)		
VI	GUI Programming-II (JavaFX)	JavaFX Basic Concepts, JavaFX application skeleton, Compiling and running JavaFX program,Simple JavaFX control:Label,Using Buttons and events, Drawing directly on Canvas. (Perform any one program that contains the concept of JavaFX). 1)Write a Java program to design a Login Form using JavaFX Controls. 2)Write Java program to draw various shapes on Canvas using JavaFX.	**04**	LO1 LO5 LO6

Text Books:

1. Herbert Schildt, "Java-The Complete Reference", Tenth Edition, Oracle Press, Tata McGraw Hill Education.
2. E. Balguruswamy, "Programming with Java A primer", Fifth edition, Tata McGraw Hill Publication
3. Anita Seth, B.L.Juneja, " Java One Step Ahead", oxford university press.

References:

1. D.T. Editorial Services, "Java 8 Programming Black Book", Dreamtech Press.

2. Learn to Master Java by Star EDU Solutions

3. Yashvant Kanetkar, "Let Us Java" ,4th Edition ,BPB Publications.

Term Work:

The Term work shall consist of at least 15 practical based on the above list. The term work Journal must include at least 2 Programming assignments. The Programming assignments should be based on real world applications which cover concepts from more than one modules of syllabus**.**

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

Practical & Oral Exam: An Oral & Practical exam will be held based on the above syllabus.

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
ITM301	Mini Project – 1 A for Front end /backend Application using JAVA	--	04	--	--	02	--	02

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Pract. /Oral	Total
		Internal assessment			End Sem. Exam			
		Test1	Test 2	Avg.				
ITM301	Mini Project – 1 A for Front end /backend Application using JAVA	--	--	--	--	25	25	50

Course Objectives

1. To acquaint with the process of identifying the needs and converting it into the problem.
2. To familiarize the process of solving the problem in a group.
3. To acquaint with the process of applying basic engineering fundamentals to attempt solutions to the problems.
4. To inculcate the process of self-learning and research.

Course Outcome: Learner will be able to…

1. Identify problems based on societal /research needs.
2. Apply Knowledge and skill to solve societal problems in a group.
3. Develop interpersonal skills to work as member of a group or leader.
4. Draw the proper inferences from available results through theoretical/ experimental/simulations.
5. Analyse the impact of solutions in societal and environmental context for sustainable development.
6. Use standard norms of engineering practices
7. Excel in written and oral communication.
8. Demonstrate capabilities of self-learning in a group, which leads to life long learning.
9. Demonstrate project management principles during project work.

Guidelines for Mini Project

Students shall form a group of 3 to 4 students, while forming a group shall not be allowed less than three or more than four students, as it is a group activity.

Students should do survey and identify needs, which shall be converted into problem statement for mini project in consultation with faculty supervisor/head of department/internal committee of faculties.

Students hall submit implementation plan in the form of Gantt/PERT/CPM chart, which will cover weekly activity of mini project.

A log book to be prepared by each group, wherein group can record weekly work progress, guide/supervisor can verify and record notes/comments.

Faculty supervisor may give inputs to students during mini project activity; however, focus shall be on self-learning.

Students in a group shall understand problem effectively, propose multiple solution and select best possible solution in consultation with guide/ supervisor.

Students shall convert the best solution into working model using various components of their domain areas and demonstrate.

The solution to be validated with proper justification and report to be compiled in standard format of University of Mumbai.

With the focus on the self-learning, innovation, addressing societal problems and entrepreneurship quality development within the students through the Mini Projects, it is preferable that a single project of appropriate level and quality to be carried out in two semesters by all the groups of the students. i.e. Mini Project 1 in semester III and IV. Similarly, Mini Project 2 in semesters V and VI.

However, based on the individual students or group capability, with the mentor's recommendations, if the proposed Mini Project adhering to the qualitative aspects mentioned above gets completed in odd semester, then that group can be allowed to work on the extension of the Mini Project with suitable improvements/modifications or a completely new project idea in even semester. This policy can be adopted on case by case basis.

Guidelines for Assessment of Mini Project:

Term Work

The review/ progress monitoring committee shall be constituted by head of departments of each institute. The progress of 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 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 log book : 10
- Marks awarded by review committee : 10
- Quality of Project report : 05

Review/progress monitoring committee may consider following points for assessment based on either one year or half year project as mentioned in general guidelines.

One-year project:

In first semester entire theoretical solution shall be ready, including components/system selection and cost analysis. Two reviews will be conducted based on presentation given by students group.

- First shall be for finalisation of problem
- Second shall be on finalisation of proposed solution of problem.

In second semester expected work shall be procurement of component's/systems, building of working prototype, testing and validation of results based on work completed in an earlier semester.

- First review is based on readiness of building working prototype to be conducted.
- Second review shall be based on poster presentation cum demonstration of working model in last month of the said semester.

Half-year project:

In this case in one semester 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 finalisation of problem and proposed solution
- Second shall be for implementation and testing of solution.

Assessment criteria of Mini Project.

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
4. Feasibility of proposed problem solutions and selection of best solution
5. Cost effectiveness
6. Societal impact
7. Innovativeness
8. Cost effectiveness and Societal impact
9. Full functioning of working model as per stated requirements
10. Effective use of skill sets
11. Effective use of standard engineering norms
12. Contribution of an individual's as member or leader
13. Clarity in written and oral communication

In **one year, project**, first semester evaluation may be based on first six criteria's and remaining may be used for second semester evaluation of performance of students in mini project.

In case of **half year project** all criteria's in generic may be considered for evaluation of performance of students in mini project.

Guidelines for Assessment of 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.

Mini Project shall be assessed based on following points;

1. Quality of problem and Clarity
2. Innovativeness in solutions
3. Cost effectiveness and Societal impact
4. Full functioning of working model as per stated requirements
5. Effective use of skill sets
6. Effective use of standard engineering norms
7. Contribution of an individual's as member or leader
8. Clarity in written and oral communication