Vivekanand Education Society's

Department of Information Technology

(Affiliated to University of Mumbai, Approved by AICTE & Recognized by Govt. of Maharashtra)

Department of Information Technology

Academic Year 2023-24

Semester III and IV

Program Structure for Second Year Information Technology

Scheme for Autonomous Program

(With Effect from 2023-2024)

Semester III

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract	Tut	Theory	TW/ Pract	Tut	Total
ITC301	Engineering Mathematics III	3	-	1	5	-	1	4
ITC302	Data Structures and Analysis	3	-	-	3	-	-	3
ITC303	Database Management System	3	-	-	3	-	-	3
ITC304	Principle of Communications	3	-	-	3	-	-	3
ITL305	Paradigm and computer programming fundamentals	3			3			3
ITL301	Data Structures Lab	-	2	-	-	1	-	1
IT302	SQL Lab	-	2	-		1	-	1
ITL303	Computer Programming paradigms Lab	-	2*	-	-	1	-	1
ITL304	Java lab(SBL)	-	4			2		2
ITM301	Mini Project -1 backend front end using JAVA	-	4			2		2
	Total	**15**	**14**	**1**	**15**	**7**	1	**23**

Course Code	Course Name	Examination Scheme							
		Theory					TW	Oral & Pract	Total
		Internal Assessment			End Sem. Exam	Exam Duration (in Hrs)			
		Mid-Test (MT)	Continuous Assessment (CA)	Total					
ITC301	Engineering Mathematics III	20	20	40	60	2	-	-	100
ITC302	Data Structures and Analysis	20	20	40	60	2	-	-	100
ITC303	Database Management System	20	20	40	60	2	-	-	100
ITC304	Principle of Communications	20	20	40	60	2	--	-	100
ITC305	Paradigm and computer programming fundamentals	20	20	40	60	2			100
ITL301	Data structures Lab	-	-	-	-	-	25	25	50
ITL302	SQL Lab	-	-	-	-	-	25	25	50
IT303	Computer programming paradigm Lab	-	-	-	-	-	25	25	50
ITL304	Java Programming Lab	-	-	-	-	-	25	25	50
ITM 301	Mini project I (JAVA)						25	25	50
Total		**100**	**100**	**200**	**300**	**-**	**125**	**125**	**750**

Course Code:	**Course Title**	**Credit**
IT301	**Engineering Mathematics III**	**4**
1)Prerequisite: Engineering Mathematics-I, Engineering Mathematics-II		
2)Course Objectives:		
1	To build a strong foundation in mathematics, provide students with mathematics fundamentals necessary to formulate, solve and analyses complex engineering problems.	
2	To prepare student to apply reasoning informed by the contextual knowledge to engineering practice, to work as part of teams on multi-disciplinary projects.	
3	To describe the ideas of Fourier and Laplace transforms and illustrate their application in the fields of PDE, Digital Signal Processing, Image Processing, Image Processing, Theory of wave equations, Differential equations, and many others.	
4	To prepare the students to use the information from Laplace transform to convert a continuous signal from the time domain to the frequency domain.	
5	To prepare the students for transforming a problem with inconvenient geometry into a one with appropriate geometry by the use of Complex mapping.	
3)Course Outcomes:		
1.	Laplace transform: . Students will be able to apply Laplace transform and its properties to find the transform of a given function and evaluate some integrals of real value function.	
2	Inverse Laplace transform : Students will be capable of solving ordinary differential equations using Laplace transform as well as problems based on determining the inverse Laplace transform of specified functions.	

3	Fourier Series : Students will be able to expand a periodic function as a Fourier series in terms of sine and cosine functions.
4	Complex Variable: Students will be proficient to construct an analytic function, obtain a family of orthogonal trajectories.
5	Complex Integration : Students will be able to evaluate integration of complex variable functions using the knowledge of Cauchy integral formula, residue of singular points.
6	Z-transform : Students will be able to find Z-transform of sequences using Properties and Inverse Z-transform using series expansion, partial fraction.

4) Syllabus

Module	**Content**	**Hrs**
Module 1	Laplace Transform: 1.1 Definition and Condition of Existence of Laplace transform. 1.2 Laplace transform of standard functions like e at , sin(at), cos(at), sinh(at), cosh(at) and t n , n ≥ 0. 1.3 Properties of Laplace transform: Linearity, First Shifting, Second Shifting, Change of Scale, Multiplication by t, Division by t, Laplace Transform of derivative, integral and convolution of two functions. 1.4 Evaluation of real improper integrals using Laplace transformation. 1.5 Laplace transform of some special functions: Heaviside's Unit Step function, Dirac Delta function.	7
Module 2	Inverse Laplace Transform: 2.1 Definition and Inverse Laplace transform of standard functions. 2.2 Inverse Laplace transform using Partial fractions, derivatives property. 2.3 Inverse Laplace transform using Convolution property. 2.4 Applications to solve initial and boundary value problems involving Ordinary differential equations.	7
Module 3	Fourier Series: 3.1 Drichlet's conditions, Definition of Fourier series and Parseval's Identity. 3.2 Fourier series of periodic function with period 2π and $2L$ 3.3 Fourier series of even and odd functions. 3.4 Half range Sine and Cosine Series.	7

Module 4	Complex Variables: 4.1 Function of complex variable f(z), Limit, Continuity and Differentiability of f(z), Analytic function. Necessary and sufficient conditions for f(z) to be Analytic. Cauchy-Riemann equations in Cartesian coordinate. 4.2 Milne-Thomson method: Determine analytic function f(z) when real part (u), imaginary part (v) or its combination is given. 4.3 Harmonic function, Harmonic conjugate and Orthogonal trajectories.	6
Module 5	Complex Integration: 5.1 Line Integral, Cauchy's Integral theorem for simple connected and multiply connected regions, Cauchy's Integral formula. 5.2 Taylor's and Laurent's series expansion. 5.3 Definition of Singularity, Zeroes, Poles of f(z), Residues, Cauchy's Residue Theorem.	7
Module 6	Z-Transform: 6.1 Definition and Region of Convergence, Transform of Standard Functions: {k n a k }, {a |k| }, { C. a k n k+n }, {c k sin(αk + β)}, {c k sinh αk}, {c k cosh αk}. 6.2 Properties of Z-Transform: Change of Scale, Shifting Property, 5 Multiplication, and Division by k, Convolution theorem. 6.3 Inverse Z-Transform: Partial Fraction Method, Convolution Method.	5
	Total	48

5) Textbooks:	
1	Higher Engineering Mathematics, Dr. B. S. Grewal, Khanna Publication
2	Linear Algebra and its Applications, D. C. Lay, Pearson
3	J.L Schiff , The Laplace Transform, Springer (1999)
6) Reference Books:	
1	J H Mathews and R W Howell, Complex Analysis for Mathematics and Engineering, Narosa
2	Advanced Engineering Mathematics, R. K. Jain and S. R. K. Iyengar, Narosa Publication.
3	Advanced Engineering Mathematics, Erwin Kreyszig, Wiley Eastern Limited
4	Complex Variables and Applications, Brown and Churchill, McGraw-Hill Education

7) Internal Assessment:
Assessment consists of one)Mid Term Test of 20 marks and Continuous Assessment of 20 marks.(Total 40
Mid Term test is to be conducted when approx. 50% syllabus is completed Duration of the midterm test shall be one hour.

8) Continuous Assessment:-
Continuous Assessment is of 20 marks. The rubrics for assessment will be considered on approval by the subject teachers. The rubrics can be any 2 or max 4 of the following:-

*For sr.no.1, the date of certification exam should be within the term and in case a student is unable to complete the certification , the grading has to be done accordingly.

9)Rubrics for slow learners:-

1.) Case study, Presentation, group discussion, technical debate on recent trends in the said course (10 marks)

2. Project based Learning and evaluation / Extra assignment / Question paper solution (10 marks)

3) Multiple Choice Questions (Quiz) (5marks)

4) Literature review of papers/journals (5 marks)

Course Code: ITC302	**Course Title: Data Structure and Analysis**	**Credit**
ITC302	**Course Title: Data Structure and Analysis**	3
Prerequisite: Introduction of C programming language.		
2) 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, queues, linked lists 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.	
3) Course Outcomes: On successful completion, of course, learner/student will be able to:		
1	Classify and Apply the concepts of stacks, queues and linked lists in real life problem solving.	
2	Classify, apply and analyze the concepts trees in real life problem solving.	
3	Illustrate and justify the concepts of graphs in real life problem solving.	

4	List and examine the concepts of searching techniques in real life problem solving.
5	List and examine the concepts of sorting techniques in real life problem solving.
6	Examine and justify different operations of stacks, queues, linked list, trees and graphs to various applications

5) Library related work (5 marks)

6)Rubrics for Indirect Assessment :-

1. Mock Viva/Practical

2. Skill Enhancement Lecture

3. Extra Assignments/lab/lecture

11)End Semester Theory Examination:	
1	Question paper will be of 60 marks
2	Question paper will comprise a total of five questions
3	All question carry 20 marks
4	Any three questions out of five needs to be solved.

4) Syllabus

Module		**Content**	**Hrs**
Module 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.. **Recursion**: Writing a recursive function, Flow of control in recursive functions, Winding and unwinding phase, **Time and space complexity of algorithm.**	**03**

Module 1	**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.	**09**
Module 2	**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	**06**

Module 3	**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. Traversals: Breadth First Search, Depth First Search. **Self-learning Topics**: Implementation of BFS, DFS	**07**
Module 4	**Searching Techniques**	Searching: Sequential Search, Binary Search. Hashing: Hash Functions: Truncation, Mid-square Method, Folding Method, Division Method. Collision Resolution: Open Addressing: Linear Probing, Quadratic Probing, Analysis of all searching techniques **Self-learning Topics: Double Hashing, Separate Chaining Bucket Hashing.**	**04**
Module 5	**Sorting Techniques**	Insertion sort, Selection sort, Merge sort, Quick sort and Radix sort. Analysis of all sorting techniques. **Self-learning Topics: Radix-Exchange Sort,shell sort**	**04**

Module 6	**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 parentheses, 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.	**06**
		Total	**39**

5) Textbooks:	
1	K Srivastava, Deepali Srivastava; Data Structures through C in Depth; BPB Publications; 2011
2	eYedidya Langsam, Moshej Augenstein, Aaron M. Tenenbaum; Data Structure Using C & C++; entice Hall of India; 1996.
3	Reema Thareja; Data Structures using C; Oxford
6) Reference Books:	
1	Ellis Horowitz, Sartaj Sahni; Fundamentals of Data Structures; Galgotia Publications; 2010 Jean Paul Tremblay, Paul G. Sorenson

2	Rakesh K. Shukla; Data Structures using C and C++; Wiley India; 2009.

7) Internal Assessment:

Assessment consists of one)Mid Term Test of 20 marks and Continuous Assessment of 20 marks.(Total 40

Mid Term test is to be conducted when approx. 50% syllabus is completed Duration of the midterm test shall be one hour.

8) Continuous Assessment:

Continuous Assessment **is of 20 marks.** The rubrics for assessment will be considered on approval by the subject teachers. The rubrics can be any 2 or max 4 of the following:-

Sr.no	Rubrics	Marks
1.	***Certificate course for 4 weeks or more:-** **NPTEL/ Coursera/ Udemy/any MOOC**	10 mark
2.	GATE Based Assignment test/Tutorials etc	10 marks
3.	Participation in event/workshop/talk / competition followed by small report and certificate of participation relevant to the subject(in other institutes)	5 marks
4.	Multiple Choice Questions (Quiz)	5 marks

* Rubrics 1 compulsory, along with rubrics rubrics 2 or (rubrics 3 & 4) students can select.

* For sr.no.1, the date of the certification exam should be within the term and in case a student is unable to complete the certification , the grading has to be done accordingly.

9) Rubrics for slow learners:-

1.) Case study, Presentation, group discussion, technical debate on recent trends in the said course (10 marks)

2) Project based Learning and evaluation / Extra assignment / Question paper solution (10 marks)

3) Multiple Choice Questions (Quiz) (5 marks)

4) Literature review of papers/journals (5 marks)

5) Library related work (5 marks)

10) Rubrics for Indirect Assessment :-

1. Mock Viva/Practical

2. Skill Enhancement Lecture

3. Extra Assignments/lab/lecture

11) End Semester Theory Examination:	
1	Question paper will be of 60 marks
2	Question paper will comprise a total of five questions
3	All question carry 20 marks
4	Any three questions out of five need to be solved.

Sample Template for Lab Work

Lab Code	Lab Name	Credit
ITL301	**Data Structure Lab**	**1**

1)Prerequisite: Introduction of C programming language.

2) Lab Objectives: The course 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.

3) Lab Outcomes: On successful completion, of course, learner/student will be able to:

1	Describe various types of Data Structures viz. stack, queue, linked list, trees, graphs; and the operations that can be performed on each. and calculate time and Space complexity of each operation
2	Describe various searching and sorting methods.

3	Select appropriate data structure to be used for the given problem
4	Explain and use the concept of recursion
5	Develop an algorithm and a C program for the given application.

4)Suggested Experiments: (minimum number of experiments to be completed can be specified)	
Sr. No.	**Name of the Experiment**
1*	Implementation of Stack Data Structure using array.
2*	Conversion of Infix Expression to Postfix Expression Using Stack
3*	Implementation of Linear Queue Data Structure using array.
4	Implementation of Circular Queue Data Structure using array.
5*	Implementation of Singly Linked List.
6*	**Linked List implementation of Stack/queue in real life application.**
7*	Implementation of Circular Singly Linked List.
8*	Implementation of Circular Doubly Linked List.

9*	Implementation of Binary Search Tree .
10*	Implementation of AVL tree.
11*	Implementation of BFS and DFS on a directed graph using an adjacency matrix.
12*	**Implementation of Binary Search in real life application**
13*	Implementation of Menu driven Selection sort, Bubble sort, Insertion sort
14*	Implementation of Menu driven Merge Sort and Quick Sort

5) Useful Links:	
1	https://www.nptel.ac.in
2	https://opendatastructures.org/
3	https://www.coursera.org/
4	https://programiz.com/dsa/

6) Term Work:

1	Term Work shall consist of at least 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)
7) Continuous assessment exam	
1	Experiment submission on time
2	Explanation/concepts
3	Algorithm implementation
4	Analysis
5	Performance/Documentation

Course Code: ITC303	**Course Title :Database Management System**	**Credit**
Currently same	**Database Management System**	3
1)Prerequisite: C Programming		
2)Course Objectives: The course aims:		
1	To learn the basics and understand the need of a database management system.	
2	To construct 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.	
3)Course Outcomes: On successful completion, of course, learner/student will be able to:		
1	Identify the need of Database Management System.	
2	Design conceptual model for real life applications.	
3	Create Relational Model for real life applications	
4	Formulate query using SQL commands.	
5	Apply the concept of normalization to relational database design.	
6	Demonstrate the concept of transaction, concurrency and recovery.	

4) syllabus

Module		**Content**	**Hrs**
Module 1	**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
Module 2	**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
Module 3	**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**
Module 4	**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. **Introduction to Query Processing and Optimization.** Self-learning Topics: Physical design of database for the relational model designed in module III and fire various queries.	08

Module 5	**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),**4NF,5NF.** **Self-learning Topics:** Consider any real time application and normalization upto 4NF/5NF	07
Module 6	**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: **Failure Classification, Storage structure, Recovery & atomicity,** Log based recovery,**Shadow paging** **Self-learning Topics:** Study the various deadlock situation which may occur for a database designed in module V.	09
		Total	**39**

5) Textbooks:	
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
6) Reference Books:	
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

7) Internal Assessment:
Assessment consists of one)Mid Term Test of 20 marks and Continuous Assessment of 20 marks.(Total 40
Mid Term test is to be conducted when approx. 50% syllabus is completed Duration of the midterm test shall be one hour.

8) Continuous Assessment:-
Continuous Assessment **is of 20 marks.** The rubrics for assessment will be considered on approval by the subject teachers. The rubrics can be any 2 or max 4 of the following:-

Sr.no	**Rubrics**	**Marks**
1.	*Certificate course for 4 weeks or more:- NPTEL/ Coursera/ Udemy/any MOOC	10 marks
2	Mini Project / Extra Experiments/ Virtual Lab	10 marks
3.	GATE Based Assignment test/Tutorials etc	10 marks
4.	Multiple Choice Questions (Quiz)	5 marks

*****For sr.no.1, the date of the certification exam should be within the term and in case a student is unable to complete the certification , the grading has to be done accordingly.

9)Rubrics for slow learners:-

1.) Case study, Presentation, group discussion, technical debate on recent trends in the said course (10 marks)

2. Project based Learning and evaluation / Extra assignment / Question paper solution (10 marks)

3) Multiple Choice Questions (Quiz) (5marks)

4) Literature review of papers/journals (5 marks)

5) Library related work (5 marks)

10)Rubrics for Indirect Assessment :-

1. Mock Viva/Practical
2. Skill Enhancement Lecture
3. Extra Assignments/lab/lecture

11)End Semester Theory Examination:	
1	Question paper will be of 60 marks
2	Question paper will comprise a total of five questions
3	All question carry 20 marks
4	Any three questions out of five need to be solved.

Sample Template for Lab Work

Lab Code	**Lab Name**	**Credit**
ITL302	SQL Lab	**1**
1)Prerequisite: The Lab experiments aims:		
2)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.	
3)Lab Outcomes: On successful completion, of course, learner/student will be able to:		
1	Define problem statement and Construct the conceptual model for real life application.	
2	Create and populate a RDBMS using SQL.	
3	Formulate and write SQL queries for efficient information retrieval	

4	Apply view, triggers and procedures to demonstrate specific event handling.
5	Demonstrate database connectivity using JDBC.
6	Demonstrate the concept of concurrent transactions.

4)Suggested Experiments: (minimum number of experiments to be completed can be specified)	
Sr. No.	**Name of the Experiment**
1	Identify real world problems and develop the problem statement. Design an Entity-Relationship (ER) / Extended Entity-Relationship (EER) Model.
2	Mapping ER/EER to Relational schema model.
3	Create a database using DDL and apply integrity constraints.
4	Perform data manipulations operations on populated databases.
5	Perform Authorization using Grant and Revoke.
6	Implement Basic and complex SQL queries.
7	Implementation of Views and Triggers.
8	Demonstrate database connectivity using JDBC.
9	Execute TCL commands.
10	Implement functions and procedures in SQL
11	Implementation of Cursor.
12	Implementation and demonstration of Transaction and Concurrency control techniques using locks.

5)Useful Links:	
1	https://www.geeksforgeeks.org/dbms/
2	https://www.w3schools.com/sql/sql_having.asp

3	https://www.tutorialspoint.com/dbms/index.htm
4	https://onlinecourses.nptel.ac.in/noc21_cs58/preview

6) Term Work:	
1	Term Work shall consist of at least 10 Practical based on the above list, but not limited to. 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)

7) Continuous assessment exam:	
1.	Timely Submission of Experiments weekwise
2.	Explanation/concept:
3.	Algorithm/implementation:
4.	Analysis
5.	Documentation/Performance:

Course Code: ITC305	**Course Title : Principle of Communication**	**Credit**
Currently same	Principle of Communication	**3**
1)Prerequisite: C Programming		
2)Course Objectives: The course aims:		
1	Study the basic principles and techniques used in analog and digital communications.	
2	Understand the concept of noise and Fourier transform for designing and analysing communication system.	
3	Acquire the knowledge of different modulation techniques such as AM , FM and study the block diagrams of transmitter and receiver.	
4	Study the Sampling theorem and Pulse Analog Modulation techniques.	
5	Learn the concepts of Digital modulation techniques such as PCM, DM, ADM and multiplexing techniques.	
6	Gain the core idea of Electromagnetic Radiation and propagation of waves.	
3)Course Outcomes: On successful completion, of course, learner/student will be able to:		
1	Describe analog and digital communication systems	
2	Differentiate types of noise, analyses the Fourier transform of time and frequency domain	
3	Design transmitter and receiver of AM, DSB, SSB and FM.	
4	Describe Sampling theorem and pulse modulation systems.	
5	Explain multiplexing and digital band pass modulation techniques.	
6	Describe electromagnetic radiation and propagation of waves.	

4) syllabus

Module	Topic	Content	Hrs
Module 1	Introduction	1.1 Basics of analog communication and digital communication systems (Block diagram). 1.2 Electromagnetic Spectrum and application, Types of Communication channels.	03
Module 2	Noise	~~2.1 Introduction to Fourier Transform, its properties (time and frequency shifting)~~ Types of Noise, Noise parameters –Signal to noise ratio, Noise factor, Noise figure. 2.2 Friss formula and Equivalent noise temperature.	05
Module 3	Amplitude and Angle modulation Techniques.	3.1 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. 3.2 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. 3.2 **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. 3.4 FM demodulator: Foster Seeley discriminator, Ratio detector	14
Module 4	Pulse Analog Modulation and Digital Modulation	4.1 Sampling theorem for low pass and band pass signals ~~with proof~~, Anti- aliasing filter, PAM, PWM and PPM generation and Degeneration.	09

		4.2 Quantization process, Pulse code modulation, Delta modulation, Adaptive delta modulation. ~~Introduction to Line codes and ISI.~~	
Module 5	Multiplexing and Digital Band Pass Modulation Techniques	5.1 Principle of Time Division Multiplexing, Frequency Division Multiplexing , and its applications 5.2 ASK, FSK, PSK, ~~QPSK~~ Generation and detection.	04
Module 6	Radiation and Propagation of Waves	6.1 Electromagnetic radiation, fundamentals, types of propagation 6.2 Ground wave, sky wave, space wave tropospheric scatter propagation.	04
		Total	**39**

	5) Textbooks:
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
	Reference Books
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.

7) Internal Assessment(20marks):
Consisting of Two Compulsory Class Tests
Approximately 40% to 50% of syllabus content must be covered in First test and remaining 40% to 50% of syllabus contents must be covered in second test.

Sr.no	Rubrics	Marks
1.	*Certificate course for 4 weeks or more:- NPTEL/ Coursera/ Udemy/any MOOC	10 marks
2	Mini Project / Extra Experiments/ Virtual Lab	10 marks
3.	GATE Based Assignment test/Tutorials etc	10 marks
4.	Multiple Choice Questions (Quiz)	5 marks

*For sr.no.1, the date of the certification exam should be within the term and in case a student is unable to complete the certification , the grading has to be done accordingly.

9)Rubrics for slow learners:-
1.) Case study, Presentation, group discussion, technical debate on recent trends in the said course (10 marks)
2. Project based Learning and evaluation / Extra assignment / Question paper solution (10 marks)
3) Multiple Choice Questions (Quiz) (5marks)
4) Literature review of papers/journals (5 marks)
5) Library related work (5 marks)

10)Rubrics for Indirect Assessment :-
1. Mock Viva/Practical
2. Skill Enhancement Lecture
3. Extra Assignments/lab/lecture

11)End Semester Theory Examination:	
1	Weightage of each module in end semester examination is expected to be/will be proportional to number of respective lecture hours mentioned in the syllabus.
2	Question paper will comprise of total six questions, each carrying 20 marks.
3	Q.1 will be compulsory and should cover maximum contents of the syllabus.
4	Remaining question will be mixed in nature (for example if Q.2 has part (a) from module 3 then part (b) will be from any other module. (Randomly selected from all the modules.)
5	Total four questions need to be solved.

Course Code: ITL304	**Course Title : Java Lab (SBL)**	**Credit**
Currently same	Java Lab (SBL)	2
1)Prerequisite: Basics of Computer Programming		
2)Course Objectives: The course 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 of inheritance, 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.	
3)Course Outcomes: On successful completion, of course, learner/student will be able to:		
1	Explain the fundamental concepts of Java Programing.	
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.	
3	Demonstrate how to extend java classes and achieve reusability using Inheritance, Interface and Packages.	
4	Construct robust and faster programmed solutions to problems using concept of Multithreading, exceptions and file handling	

5	Design and develop Graphical User Interface using Abstract Window Toolkit and Swings along with response to the events.
6	Develop Graphical User Interface by exploring JavaFX framework based on MVC architecture.

4) syllabus

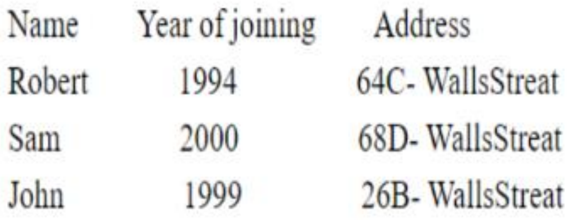
Module		**Content**	**Hrs**
Module 1	**Java Fundamentals**	**Overview of procedure and object oriented Programming, Java Designing Goals and Features of Java Language**. **Java virtual machine** **Introduction to the principles of object-oriented programming:** Classes, Objects, Abstraction, Encapsulation, 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 **Java Collection** (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.	**07**

		3) Write a Java program that prints all real solutions to the quadratic equation ax2+bx+c = 0. Read in a, b, c and use the quadratic formula. If the discriminate b2-4ac 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 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.	
Module 2	**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. **Java Data Structures**	**07**

(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.

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.	
Module 3	**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. 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.	**10**

		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.	
Module 4	**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. **Multithreading advantages and issues.** (Perform any 3 programs that cover Exception Handling, Multithreading and I/O Streams). **Experiments:** 1) Write java program where user will enter login id 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	**10**

		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 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	
Module 5	**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 Model of Event Handling, Event Classes, Event Sources, Event Listeners, Adapter Classes as Helper Classes in Event Handling.	**12**

		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. 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.)	
Module 6	**GUI Programming-II (JavaFX) Java Applets**	JavaFX Basic Concepts, JavaFX application skeleton, Compiling and running JavaFX program,Simple JavaFX control:Label,Using Buttons and events, Drawing directly on Canvas.	**04**

		(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. **Applet Programming:** local and remote applets, difference between applet and application, applet life cycle, developing executable applet code. 1)Develop code for simple Java applets.	
		Total	**50**

	5) Textbooks:
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. Korth, Slberchatz, Sudarshan, Database System Concepts, 6th Edition, McGraw Hill
	6) Reference Books:
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.

7) 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)

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

9)Rubrics for slow learners:-

1.) Case study, Presentation, group discussion, technical debate on recent trends in the said course (10 marks)

2. Project based Learning and evaluation / Extra assignment / Question paper solution (10 marks)

3) Multiple Choice Questions (Quiz) (5marks)

4) Literature review of papers/journals (5 marks)

5) Library related work (5 marks)

10)Rubrics for Indirect Assessment :-

1. Mock Viva/Practical

2. Skill Enhancement Lecture

3. Extra Assignments/lab/lecture

Course Code: ITM301	**Course Title :** **Mini Project – 1 A for Front end /backend Application using JAVA**	**Credit**
Currently same	**Mini Project – 1 A for Front end /backend Application using JAVA**	2
1)Prerequisite:		
2)Course Objectives: The course aims:		
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.	
3)Course Outcomes: On successful completion, of course, learner/student 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.

4) 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 nfaculties.
- 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.

5) 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

6) 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.

7) 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.

8) 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

Program Structure for Second Year Information Technology

Scheme for Autonomous Program

(With Effect from 2023-2024)

Semester IV

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract	Tut	Theory	TW/ Pract	Tut	Total
ITC401	Engineering Mathematics –IV	3	-	1	5	-	1	4
ITC402	Computer Network and Network Design	3	-	-	3	-	-	3
ITC403	Operating System	3	-	-	3	-	-	3
ITC404	Automata Theory	3	-	-	3	-	-	3
ITC405	Computer Organization and Architecture	3			3			3
ITL401	Network Lab	-	2	-	-	1	-	1
ITL402	Unix Lab	-	2	-		1	-	1
ITL403	Microprocessor Lab	-	2	-	-	1	-	1
ITL404	Python Lab (SBL)	-	4			2		2
ITM401	Mini Project – 1 B for Python Based automation projects	-	4			2		2
	Total	**15**	**14**	**1**	**15**	**7**	**1**	**23**

Course Code	Course Name	Examination Scheme							
		Theory					TW	Oral & Pract	Total
		Internal Assessment			End Sem. Exam	Exam Duration (in Hrs)			
		Mid-Test (MT)	Continuous Assessment (CA)	Total					
ITC401	Engineering Mathematics –IV	20	20	40	60	2	-	-	100
ITC402	Computer Network and Network Design	20	20	40	60	2	-	-	100
ITC403	Operating System	20	20	40	60	2	-	-	100
ITC404	Automata Theory	20	20	40	60	2	-	-	100
ITC405	Computer Organization and Architecture	20	20	40	60	2			100
ITL401	Network Lab	-	-	-	-	-	25	25	50
ITL402	Unix Lab	-	-	-	-	-	25	25	50
ITL403	Microprocessor Lab	-	-	-	-	-	25	25	50
ITL404	Python Lab (SBL)	-	-	-	-	-	25	25	50
ITM401	Mini Project – 1 B for Python Based automation projects						25	25	50
Total		**100**	**100**	**200**	**300**	**-**	**125**	**125**	**750**

@ 4 hours shown as theory to be taken class wise and 1 hour to be taken tutorial as class wise

$ 3 hours shown as theory to be taken class wise and 1 hour to be taken tutorial as batch wise

* 2 hours shown as practical’s to be taken class wise lecture and another 2 hours to be taken as batch wise practices in the lab.

Course Code:	**Course Title**	**Credit**
IT 401	**Applied Mathematics IV**	**4**
1)Prerequisite: Engineering Mathematics-I, Engineering Mathematics-II, Engineering Mathematics-III		
2)Course Objectives:		
1	To build a strong foundation in mathematics, provide students with mathematics fundamentals necessary to formulate, solve and analyses complex engineering problems. To prepare student to apply reasoning informed by the contextual knowledge to engineering practice, to work as part of teams on multi-disciplinary projects.	
2	To prepare student to apply reasoning informed by the contextual knowledge to engineering practice, to work as part of teams on multi-disciplinary projects.	
3	To acknowledge the importance of sampling design and analysis methods for research and management in many other fields.	
4	To get familiar with the mathematical formulation of a real world problem, acquaint with the problem solving techniques theoretically, tackle several parameters into account while dealing with the problem and make aware the students about the applications of various forms of Linear Programming.	
5	To prepare students to apply linear algebra concepts to model, solve and analyse real-world situations.	
6	To prepare students to apply the concept of eigenvalues and Eigen vector which will further be useful in applications like Google page rank algorithms, principal component analysis (biometric systems), and natural frequency for a structure.	
7	To prepare the students to use a powerful statistical software platform SPSS (Statistical Package for the Social Sciences) for the analysis of statistical data in the future.	
3)Course Outcomes:		
1	Probability theory: Students will understand various probability measures, distribution functions, and their characteristics. They will be able to Compute probability using probability distribution of discrete and continuous Random variable.Additionally,the knowledge regarding Bayes theorem will help them take various real-life problems that arise in the medical fields and industries.	
2	Probability Distribution and Sampling Theory : Students will know fundamental concepts of testing of hypothesis, formulation of statistical hypothesis in real-life situations, developing best test procedures to test the hypothesis, and the principles underlying sampling as a means of making inferences about a population. They can also apply the idea of probability distribution to engineering problems .	

3	Statistical Techniques: Students will apply the concept of Correlation and Regression, fitting of curve to the given data sets.
4	Eigenvalues and Eigenvectors: Students will be able to execute matrix diagonalization and perform basic eigenvalue and eigenvector computations.
5	Linear Programming Problems: Students should be able to formulate a given simplified description of a suitable real-world problem as a linear programming model in general, standard and canonical forms.linear programming models can be solved by them using the simplex method , Big M method and Dual simplex method.
6	Non Linear Programming Problems: Students will be able to solve Non Linear Optimization problems using Lagrange’s multiplier method and Karush Kuhn Tucker Method.

4) syllabus

Module	Content	Hrs
Module 1	**Probability:** 2.1 Definition and basics of probability, conditional probability. Total Probability theorem and Bayes’ theorem. 2.2 Discrete and continuous random variable with probability distribution and probability density function. 2.3 Expectation, Variance, Moment generating function, Raw and central Moments, Covariance, Skewness and Kurtosis of distribution and their properties. 2.4 Probability Distribution: Binomial, Poisson and Normal distribution.	08
Module 2	**Probability Distribution and Sampling Theory:** 3.1 Sampling distribution, Test of Hypothesis, Level of Significance, Critical region, One-tailed, and two-tailed test, Test of significance of mean and difference between the means of two samples for Large samples. 3.2 Degree of freedom, Student’s t-distribution, Test of significance of mean and difference between the means of two samples for Small samples. 3.3 Chi-Square Test: Test of goodness of fit. Contingency table and Test of independence of attributes.	07
Module 3	**Statistical Techniques:** 4.1 Karl Pearson’s coefficient of correlation (r). 4.2 Spearman’s Rank correlation coefficient (R) (with repeated and non-repeated ranks). 4.3 Fitting of first and second degree curves. 4.4 Lines of regression.	5
Module 4	**Linear Algebra (Theory of Matrices):** 1.1 Characteristic Equation, Eigen values and Eigen vectors, and properties.	7

	1.2 Cayley-Hamilton Theorem ,verification and reduction of higher degree polynomials. 1.3 Similarity of matrices, diagonalizable and non-diagonalizable matrices	
Module 5	**Linear Programming Problems:** 5.1 Types of solutions, Standard and Canonical of LPP, Basic and Feasible solutions, slack variables, surplus variables, Simplex method. 5.2 Artificial variables, Big-M method (Method of penalty) 5.3 Duality, Dual of LPP and Dual Simplex Method.	6
Module 6	**Nonlinear Programming Problems:** 6.1 NLPP with no constraint, one equality constraint . using the method of Lagrange's multipliers. 6.2 NLPP with two equality constraints. 6.3 NLPP with inequality constraint: Karush-Kuhn-Tucker (KKT) conditions.	6
	Total	**46**

5) Textbooks:	
1	Advanced Engineering Mathematics, R. K. Jain and S. R. K. Iyengar, Narosa.
2	Gupta and Kapoor, Fundamental of Mathematical Statistics, S Chand
3	Operations Research, Hira and Gupta, S. Chand Publication.

6) Reference Books:	
1	Advanced Engineering Mathematics, Erwin Kreyszig, John Wiley & Sons.
2	Probability, Statistics and Random Processes, T. Veerarajan, McGraw-Hill Education.

7) Internal Assessment:

Assessment consists of one)Mid Term Test of 20 marks and Continuous Assessment of 20 marks.(Total 40
Mid Term test is to be conducted when approx. 50% syllabus is completed Duration of the midterm test shall be one hour.

8) Continuous Assessment:-

Continuous Assessment is of 20 marks. The rubrics for assessment will be considered on approval by the subject teachers. The rubrics can be any 2 or max 4 of the following:-

Sr.no	Rubrics	Marks
1.	*Certificate course for 4 weeks or more:- NPTEL/ Coursera/ Udemy/any MOOC	10 marks
2.	Wins in the event/competition/hackathon	10 marks
3.	Content beyond syllabus presentation	10 marks
4.	Creating Proof of concept	10 marks
5.	Mini Project / Extra Experiments/ Virtual Lab	10 marks
6.	GATE Based Assignment test/Tutorials etc	10 marks
7.	Participation in event/workshop/talk / competition followed by small report and certificate of participation relevant to the subject(in other institutes)	5 marks
8.	Multiple Choice Questions (Quiz)	10 marks

*For sr.no.1, the date of certification exam should be within the term and in case a student is unable to complete the certification , the grading has to be done accordingly.

9)Rubrics for slow learners:-

1.) Case study, Presentation, group discussion, technical debate on recent trends in the said course (10 marks)
2. Project based Learning and evaluation / Extra assignment / Question paper solution (10 marks)
3) Multiple Choice Questions (Quiz) (5marks)
4) Literature review of papers/journals (5 marks)
5) Library related work (5 marks)

10)Rubrics for Indirect Assessment :-

1. Mock Viva/Practical
2. Skill Enhancement Lecture
3. Extra Assignments/lab/lecture

11)End Semester Theory Examination:	
1	Question paper will be of 60 marks
2	Question paper will comprise a total of five questions
3	All question carry 20 marks
4	Any three questions out of five needs to be solved.

6)Term Work:	
1	(breakup for term work can be specified by the subject teacher)
7)Continuous assessment exam	
1	(breakup and conditions can be specified by subject teacher)

Course Code: ITC402	**Course Title: Computer Networks**	**Credit**
ITC402	**Course Title: Computer Networks**	3
2) Course Objectives: The course aims:		
1	Study the basic taxonomy and terminology of the computer networking and enumerate the layers of OSI model and TCP/IP model.	
2	Acquire knowledge of Application layer and Presentation layer paradigms and protocols.	
3	Study Session layer design issues, Transport layer services, and protocols.	
4	Gain core knowledge of Network layer routing protocols and IP addressing.	
5	Study data link layer concepts, design issues, and protocols.	
6	Read the fundamentals and basics of Physical layer, and will apply them in real time applications.	
3) Course Outcomes: On successful completion, of course, learner/student will be able to:		
1	Describe the functions of each layer in OSI and TCP/IP model.	

Course Code: ITC402	**Course Title: Computer Networks**	**Credit**
ITC402	**Course Title: Computer Networks**	3
2	Explain the functions of Application layer and Presentation layer paradigms and Protocols.	
3	Describe the Session layer design issues and Transport layer services.	
4	Classify the routing protocols and analyze how to assign the IP addresses for the given network.	
5	Describe the functions of data link layer and explain the protocols.	
6	Explain the types of transmission media with real time applications.	

4) Syllabus

Sr. No.	**Module**	**Detailed Content**	**Hours**	**CO Mapping**
0	Prerequisite	Von Neumann model, Modulation, Demodulation, encoding, Decoding.	02	--
I	Introduction	Network Criteria, Physical Structures, Network Types: LAN, WAN, Switching, OSI Reference model, TCP/IP suite, Comparison of OSI and TCP/IP, Network devices.	04	CO1

II	Application layer and Presentation layer	Introduction: Providing Services, Application layer Paradigms, Client- Server Paradigm: Application Programming Interface, Using Services of the Transport Layer, Standard Client Server applications: World Wide Web and HTTP, FTP, Electronic Mail, TELNET, Secure Shell (SSH), Domain Name System (DNS), Compression: Lossless Compression, Lossy Compression, Multimedia data: Text, Image, Video , Audio ,Multimedia in the Internet: Streaming Stored Audio/Video, Streaming Live Audio/Video, Real-Time Interactive Audio/Video, Optimal Compression Algorithms, Huffman Coding, Adaptive Huffman Compression, Dictionary Based Compression, Speech Compression, LZW, RLE, Image Compression – GIF,JPEG.	10	CO1 CO2
III	Session layer and Transport layer	Session layer design issues, Session Layer protocol - Remote Procedure Call (RPC), Transport layer services, Transport Layer Protocols: Simple Protocol, Stop-and-Wait Protocol, Go- Back-N Protocol (GBN), Selective- Repeat Protocol, Bidirectional Protocols: Piggybacking, Internet Transport-Layer Protocols, User Datagram Protocol: User Datagram, UDP Services, UDP Applications, Transmission Control Protocol: TCP Services, TCP Features, Segment, Segment, A TCP Connection, State Transition Diagram, Windows in TCP, Flow Control, Error Control, TCP Congestion Control, TCP Timers, Options.	10	CO1 CO3
IV	Network Layer	Introduction: Network-Layer Services, Packet Switching, Network-Layer Performance, Network-Layer Performance, Network-Layer Congestion, Structure of A Router, Network Layer Protocols: IPv4 Datagram Format, IPv4 Addresses,	12	CO1 CO4

		Forwarding of IP Packets, ICMPv4, Unicast Routing: General Idea, Routing Algorithms, Unicast Routing Protocols, Multicast Routing : Introduction, Multicasting Basics, Intradomain Routing Protocols, Interdomain Routing Protocols, Next generation IP: Packet Format , IPv6 Addressing , Transition from IPv4 to IPv6, ICMPv6, Mobile IP: Addressing , Agents , Three Phases , Inefficiency in Mobile IP.		
V	Data Link Layer	Wired Networks; Introduction: Nodes and Links, Two Types of Links, Two Sublayers, Data Link Control: Framing, Flow and Error Control, Error Detection and Correction, Two DLC Protocols, Medium Access Protocols: Random Access, Controlled Access, Channelization, Link Layer Addressing, Wired LANS: Ethernet Protocol; IEEE Project 802, Standard Ethernet, Fast Ethernet (100 Mbps), Gigabit Ethernet, 10-Gigabit Ethernet, Virtual LANs, Other Wired Networks: Point-to-Point Networks, SONET, Switched Network: ATM, Connecting Devices: Repeaters or Hubs, Link-Layer Switches, Routers, Sliding Window Compression.	09	CO1 CO5

VI	Physical Layer	Data and Signals: Analog and Digital, Transmission Impairment, Data Rate Limits, Performance, Digital Transmission: Digital-to-Digital Conversion , Analog-to-Digital Conversion, Analog Transmission: Digital-to-Analog Conversion, Analog- to-Analog Conversion ,Bandwidth Utilization: Multiplexing, Spread Spectrum, Transmission Media: Guided Media, Unguided Media: Wireless, Real Time Interactive Protocols: Rationale for New Protocols, RTP, Session Initialization Protocol (SIP), H.323, SCTP.	05	CO1 CO6

5) Textbooks:	
1	Behrouz A. Forouzan, Forouzan Mosharrat , Computer Networks A Top down Approach, Mc Graw Hill ıcation.
2	Andrew S Tanenbaum, Computer Networks -, 4th Edition, Pearson Education.
3	Ranjan Bose, Information Theory, Coding and Cryptography, Ranjan Bose, Tata McGrawHill , Second ition.
6) Reference Books:	
1	Behrouz A. Forouzan, Data communications and Networking, Fifth edition TMH 2013.
2	se, K. W. Ross, Computer Networking: A Top-Down Approach Featuring the Internet, 3rd Edition, Pearson Eduction.

7) Internal Assessment (20 marks):
Consisting of Two Compulsory Class Tests
Approximately 40% to 50% of syllabus content must be covered in First test and remaining 40% to 50% of syllabus contents must be covered in second test.

Sr.n	Rubrics	Marks
1.	***Certificate course for 4 weeks or more:-** **NPTEL/ Coursera/ Udemy/any MOOC**	10 marks
2.	GATE Based Assignment test/Tutorials etc	10 marks
3.	Participation in event/workshop/talk / competition followed by small report and certificate of participation relevant to the subject(in other institutes)	5 marks
4.	Multiple Choice Questions (Quiz)	5 marks

***** Rubrics 1 compulsory, along with rubrics rubrics 2 or (rubrics 3 & 4) students can select.

***** For sr.no.1, the date of the certification exam should be within the term and in case a student is unable to complete the certification , the grading has to be done accordingly.

8) Rubrics for slow learners:-

1.) Case study, Presentation, group discussion, technical debate on recent trends in the said course (10 marks)

2. Project based Learning and evaluation / Extra assignment / Question paper solution (10 marks)

3) Multiple Choice Questions (Quiz) (5 marks)

4) Literature review of papers/journals (5 marks)

5) Library related work (5 marks)

9) Rubrics for Indirect Assessment :-

1. Mock Viva/Practical

2. Skill Enhancement Lecture

3. Extra Assignments/lab/lecture

Extra Assignments/lab/lecture

10)End Semester Theory Examination:	
1	Weightage of each module in end semester examination is expected to be/will be proportional to number of respective lecture hours mentioned in the syllabus.
2	Question paper will comprise of total six questions, each carrying 20 marks.
3	Q.1 will be compulsory and should cover maximum contents of the syllabus.
4	Remaining question will be mixed in nature (for example if Q.2 has part (a) from module 3 then part (b) will be from any other module. (Randomly selected from all the modules.)
5	Total four questions need to be solved.

Course Code: ITC403	**Course Title :Operating System**	**Credit**
Currently same	**Operating System**	3
1)Prerequisite: C Programming		
2)Course Objectives: The course aims:		
1	To understand the major components of the Operating System & its functions.	
2	To introduce the concept of a process and its management like transition, scheduling, etc.	
3	To understand basic concepts related to Inter-process Communication (IPC) like 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 Systems 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.	
3)Course Outcomes: On successful completion, of course, learner/student will be able to:		
1	Understand the basic concepts related to Operating Systems.	
2	Describe the process management policies and illustrate scheduling of processes by CPU.	
3	Explain and apply synchronization primitives and evaluate deadlock conditions as handled by the Operating System.	
4	Describe and analyze the memory allocation and management functions of Operating Systems.	
5	Analyze and evaluate the services provided by the Operating System for storage management.	

6	Compare the functions of various special-purpose Operating Systems.

4) syllabus

Module		**Content**	**Hrs**
Module 1	**Fundamentals of Operating System**	Introduction to Operating Systems; Operating System Structure-**Layered,Monolithic,Microkernel operating system**; Functions of Operating Systems; Operating System Services and Interface; System Calls and its Types; System Programs; Operating System Structure. Self-learning Topics: Study of any three different OS. System calls with examples for different OS.	05
Module 2	**Process Management**	Basic Concepts of Process; Operation on Process; Process State Model and Transition; Process Control Block; Context Switching; Introduction to Threads; Types of Threads, Thread Models; Basic Concepts of Scheduling; Types of Schedulers; Scheduling Criteria; Scheduling Algorithms and **performance evaluation of the scheduling.** **Self-learning Topics:** Performance comparison of Scheduling Algorithms, Selection of Scheduling Algorithms for different situations, Real-time Scheduling	06
Module 3	**ProcessCoordination**	Basic Concepts of Inter-process Communication and Synchronization; Race Condition; Critical Region and Problem; Peterson's Solution; Synchronization Hardware and Semaphores; **Monitors**,Classic Problems of Synchronization; Message Passing; Introduction to Deadlocks; System Model, Deadlock Characterization; Deadlock Detection and Recovery; Deadlock Prevention; Deadlock Avoidance. **Self-learning Topics:** Study a real time case study for Deadlock detection and recovery.	09
Module 4	**Memory Management**	Basic Concepts of Memory Management;**Logical and Physical address map**,Swapping; Memory Allocation :**Contiguous**	09

		memory allocation-Fixed and variable partition-Internal and External fragmentation and compaction; Paging; Structure of Page Table;Segmentation ; Basic Concepts of Virtual Memory; Demand Paging, Page Replacement Algorithms; Thrashing. **Self-learning Topics:** Memory Management for any one Operating System, Implementation of Page Replacement Algorithms.	
Module 5	**Storage Management**	Basic Concepts of File System; File Access Methods; Directory Structure; File-System Implementation; Allocation Methods; Free Space Management; 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
Module 6	**Special-purpose Operating Systems**	Open-source and Proprietary Operating System; Fundamentals of Distributed Operating System; Network Operating System;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
		Total	**39**

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

7) Internal Assessment:
Assessment consists of one)Mid Term Test of 20 marks and Continuous Assessment of 20 marks.(Total 40 Mid Term test is to be conducted when approx. 50% syllabus is completed Duration of the midterm test shall be one hour.

8) Continuous Assessment:-
Continuous Assessment **is of 20 marks.** The rubrics for assessment will be considered on approval by the subject teachers. The rubrics can be any 2 or max 4 of the following:-

Sr.r	Rubrics	Marks
1.	***Certificate course for 4 weeks or more:-** **NPTEL/Coursera/Udemy/any MOOC**	**10 marks**
2.	**Extra Experiments/ Virtual Lab/GATE Based Assignment test/Tutorials etc**	**10 marks**
3.	**IA Test**	**20 Marks**

*****For sr.no.1, the date of the certification exam should be within the term and in case a student is unable to complete the certification , the grading has to be done accordingly.

9)Rubrics for slow learners:-

1.) Case study, Presentation, group discussion, technical debate on recent trends in the said course (10 marks)

2. Project based Learning and evaluation / Extra assignment / Question paper solution (10 marks)

3) Multiple Choice Questions (Quiz) (5marks)

4) Literature review of papers/journals (5 marks)

5) Library related work (5 marks)

10)Rubrics for Indirect Assessment :-

1. Mock Viva/Practical

2. Skill Enhancement Lecture

3. Extra Assignments/lab/lecture

11)End Semester Theory Examination:	
1	Question paper will be of 60 marks
2	Question paper will comprise a total of five questions
3	All question carry 20 marks
4	Any three questions out of five need to be solved.

Course Code: ITC405	**Course Title: Automata Theory**	**Credit**
ITC405	**Course Title: Automata Theory**	3
2) Course Objectives: The course aims:		
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 and machines as Acceptors, Verifiers and Translators.	
4	To understand the relation between Contexts free Languages, PDA and TM.	
5	To learn how to design PDA as acceptor and TM as Calculators.	
6	To learn how to co-relate Automata's with Programs and Functions.	
3) Course Outcomes: On successful completion, of course, learner/student will be able to:		
1	Understand, design, construct, analyze and interpret Regular languages, Expression and Grammars.	
2	Design different types of Finite Automata and Machines as Acceptor, Verifier and Translator.	
3	Understand, design, analyze and interpret Context Free languages, Expression and Grammars.	
4	Design different types of Push down Automata as Simple Parser.	
5	Design different types of Turing Machines as Acceptor, Verifier, Translator and Basic computing machine.	
6	Compare, understand and analyze different languages, grammars, Automata and Machines and appreciate their power and convert Automata to Programs and Functions.	

4) Syllabus

Sr. No.	Module	Detailed Content	Hours	CO Mapping
I	Introductio n and Regular	Languages: Alphabets and Strings. Regular Languages: Regular Expressions, Regular Languages, Regular Grammars, RL and LL	06	CO1
II	Finite Automata and machines	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: m/c with output Moore and Mealy machines. M/c as translators. Melay and Moore m/c Conversion	09	CO2
III	Context Free Grammars	Context Free Languages: CFG, Leftmost and Rightmost derivations, Ambiguity, Simplification and Normalization (CNF) and Chomskey Hierarchy (Types 0 to 3)	08	CO3
IV	Push Down Automata	Push Down Automata: Deterministic (single stack)PDA, Equivalence between PDA and CFG.	05	CO4
V	Turing Machine	Turing Machine: Deterministic TM , Multi-track and Multi-tape TMs, concept of UTM and idea of system program. Issue and concept of Halting Problem	07	CO5
VI	Application s of Automata	1. Power and Limitations of Regular and Context Free Grammars and Machines 2. Designing Functions: FA: Acceptor and Verifier. FSM: Translator PDA: Simple Parser for WF parenthesis, palindromes etc. TM: Basic bit wise calculator(+ /- /AND/OR) and Translator **(Note Added)**	04	CO2 CO4 CO5 CO6

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
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	Theory of Computation - By Vivek Kulkarni from Oxford University.

Suggested Tutorials:

Sr. No.	Module	Detailed Content
I	Introduction and Regular Languages	1 Tutorial on design of RE, RG, RLG and LLG for given Regular Language.
II	Finite Automata and machines	3 Tutorials for converting RE to NFA, NFA to DFA to Reduced DFA, FA to RE. 1 Tutorial on design of Moore and Mealy machines.
III	Context Free Grammars	1 Tutorial on design of CFG and Leftmost and Rightmost derivations. 1 Tutorial for converting CFG to CNF.
IV	Push Down Automata	1 Tutorial on design of Push Down Automata.
V	Turing Machine	1 Tutorial on design of single tape Turing Machine. 1 Tutorial on design of Multi-track and Multi-tape TMs.
VI	Applications of Automata	2 Tutorials for converting Automata to Functions: a. FA to Acceptor / Verifier. b. FSM to Translator. c. PDA to Simple Parser for WF parenthesis, palindromes etc. d. TM to Basic bit wise calculator(+ /- /AND/OR) / Translator

7) Internal Assessment (20 marks):

Consisting of Two Compulsory Class Tests

Approximately 40% to 50% of syllabus content must be covered in First test and remaining 40% to 50% of syllabus contents must be covered in second test.

Sr.r	Rubrics	Marks
1.	***Certificate course for 4 weeks or more:-** **NPTEL/ Coursera/ Udemy/any MOOC**	10 marks
2.	GATE Based Assignment test/Tutorials etc	10 marks
3.	Participation in event/workshop/talk / competition followed by small report and certificate of participation relevant to the subject(in other institutes)	5 marks
4.	Multiple Choice Questions (Quiz)	5 marks

***** Rubrics 1 compulsory, along with rubrics rubrics 2 or (rubrics 3 & 4) students can select.

***** For sr.no.1, the date of the certification exam should be within the term and in case a student is unable to complete the certification , the grading has to be done accordingly.

8) Rubrics for slow learners:-

1.) Case study, Presentation, group discussion, technical debate on recent trends in the said course (10 marks)

2. Project based Learning and evaluation / Extra assignment / Question paper solution (10 marks)

3) Multiple Choice Questions (Quiz) (5 marks)

4) Literature review of papers/journals (5 marks)

5) Library related work (5 marks)

9) Rubrics for Indirect Assessment :-

1. Mock Viva/Practical

2. Skill Enhancement Lecture

3. Extra Assignments/lab/lecture

Extra Assignments/lab/lecture

10)End Semester Theory Examination:	
1	Weightage of each module in end semester examination is expected to be/will be proportional to number of respective lecture hours mentioned in the syllabus.
2	Question paper will comprise of total six questions, each carrying 20 marks.
3	Q.1 will be compulsory and should cover maximum contents of the syllabus.
4	Remaining question will be mixed in nature (for example if Q.2 has part (a) from module 3 then part (b) will be from any other module. (Randomly selected from all the modules.)
5	Total four questions need to be solved.

Course Code: ITC405	**Course Title :Computer Organization and Architecture**	**Credit**
Currently same	**Computer Organization and Architecture**	03
1)Prerequisite: Basics of Electrical Engineering, Fundamentals of Computer		
2)Course Objectives: The course aims:		
1	Learn the fundamentals of Digital Logic Design.	
2	Conceptualize the basics of organizational and features of a digital computer.	
3	Study microprocessor architecture and assembly language programming.	
4	Study processor organization and parameters influencing performance of a processor.	
5	Analyse various algorithms used for arithmetic operations.	
6	Study the function of each element of memory hierarchy and various data transfer techniques used in digital computers.	
3)Course Outcomes: On successful completion, of course, learner/student will be able to:		
1	Demonstrate the fundamentals of Digital Logic Design	
2	Describe basic organization of computer, the architecture of 8086 microprocessor and implement assembly language programming for 8086 microprocessors.	
3	Demonstrate control unit operations and conceptualize instruction level parallelism.	
4	List and Identify integers and real numbers and perform computer arithmetic operations on integers.	

5	Categorize memory organization and explain the function of each element of a memory hierarchy.
6	Examine different methods for computer I/O mechanism.

4) syllabus

Module		**Content**	**Hrs**
Module 1	**Fundamentals of Logic Design**	Number systems: Introduction to Number systems,Binary Number systems, Signed Binary Numbers,Binary, Octal, Decimal and Hexadecimal number and their conversions, 1's and 2's complement Combinational Circuits: NOT,AND,OR,NAND,NOR,EX-OR,EX-NOR Gates. Half & Full Adder and subtractor, Reduction of Boolean functions using K-map method (2,3,4 Variable), introduction to Multiplexers and Demultiplexers, Encoders & Decoders.Sequential Circuits: Introduction to Flip Flops: SR, JK,D, T, master slave flip flop, Truth Table. **Self-learning Topics:** Number System, Quine- McCluskey,Flip-Flop conversion, Counter Design.	07
Module 2	**Overview of Computer Architecture & Organization**	Introduction of Computer Organization and Architecture. Basic organization of computer and block level description of the functional units. Evolution of Computers, Von Neumann model. Performance measure of Computer Architecture, Architecture of 8086 Family, Instruction Set, Addressing Modes, Assembler Directives, Mixed-Language Programming, Stack, Procedure, Macro. Self-learning Topics: Interfacing of I/O devices with 8086(8255,ADC,DAC).	08

Module 3	**Processor Organization and Architecture**	CPU Architecture, Instruction formats, basic instruction cycle with Interrupt processing. Instruction interpretation and sequencing. Control Unit: Soft wired (Microprogrammed) and hardwired control unit design methods. Microinstruction sequencing and execution. Introduction to parallel processing concepts, Flynn's classifications, instruction pipelining, pipeline hazards. **Self-learning Topics:** Study the examples on instruction pipelining for practice.	07
Module 4	**Data Representation and Arithmetic Algorithms**	Booth's algorithm. Division of integers: Restoring and non-restoring division, signed division, basics of floating-point representation IEEE 754 floating point (Single & double precision) number representation. **Self-learning Topics:** Implement Booth's Algorithm and Division methods.	06
Module 5	**Memory Organization**	Introduction to Memory and Memory parameters.Classifications of primary and secondary memories. Types of RAM and ROM, Allocation policies, Memory hierarchy and characteristics. Cache memory: Concept,architecture (L1, L2, L3), mapping techniques. **Self-learning Topics:** Case study on Memory Organization, Numerical on finding EAT, Address mapping.	07
Module 6	**I/O Organization**	Input/output systems, I/O module- need & functions and Types of data transfer techniques: Programmed I/O, Interrupt driven I/O and DMA.	04

		Self-learning Topics: Comparison of all I/O methods.	
		Total	**39**

5) Textbooks:	
1	R. P. Jain,”Modern Digital Electronics”, TMH
2	M. Morris Mano,”Digital Logic and Computer Design”, PHI
3	Carl Hamacher, Zvonko Vranesic and Safwat Zaky, Computer Organization, Fifth Edition, Tata McGraw-Hill.
4	William Stallings, Computer Organization and Architecture: Designing for Performance, Eighth Edition, Pearson
6) Reference Books:	
1	A. Anand Kumar, ”Fundamentals of Digital Circuits”,. PHI
2	Donald P Leach, Albert Paul Malvino, “Digital Principals & Applications”, TMH.
3	B. Govindarajulu,, Computer Architecture and Organization: Design Principles and Applications, Computer Architecture and Organization: Design Principles and Applications, Tata McGraw-Hill
4	Dr. M. Usha, T. S. Srikanth, Computer System Architecture and Organization, First Edition, Wiley-India.
5	John P. Hayes, Computer Architecture and Organization, Third Edition., McGraw-Hill

7) Internal Assessment:

Assessment consists of one)Mid Term Test of 20 marks and Continuous Assessment of 20 marks.(Total 40 Mid Term test is to be conducted when approx. 50% syllabus is completed Duration of the midterm test shall be one hour.

8) Continuous Assessment:-

Continuous Assessment **is of 20 marks.** The rubrics for assessment will be considered on approval by the subject teachers. The rubrics can be any 2 or max 4 of the following:-

Sr.r	**Rubrics**	**Marks**
1.	***Certificate course for 4 weeks or more:-** **NPTEL/Coursera/Udemy/any MOOC**	**10 mark**
2.	**Extra Experiments/ Virtual Lab /GATE Based Assignment test/Tutorials etc**	**10 marks**
3.	**IA Test**	**20 Marks**

*****For sr.no.1, the date of the certification exam should be within the term and in case a student is unable to complete the certification , the grading has to be done accordingly.

9)Rubrics for slow learners:-

1.) Case study, Presentation, group discussion, technical debate on recent trends in the said course (10 marks)

2. Project based Learning and evaluation / Extra assignment / Question paper solution (10 marks)

3) Multiple Choice Questions (Quiz) (5marks)

4) Literature review of papers/journals (5 marks)

5) Library related work (5 marks)

10)Rubrics for Indirect Assessment :-

1. Mock Viva/Practical

2. Skill Enhancement Lecture

3. Extra Assignments/lab/lecture

11)End Semester Theory Examination:	
1	Question paper will be of 60 marks
2	Question paper will comprise a total of five questions
3	All question carry 20 marks
4	Any three questions out of five need to be solved.

Sample Template for Lab Work

Lab Code	**Lab Name**	**Credit**
ITL403	**Microprocessor Lab**	**1**

1)Prerequisite: The Lab experiments aims:	
2)Lab Objectives:	
1	Learn assembling and disassembling of PC
2	Design, simulate and implement different digital circuits
3	Get hands-on experience with Assembly Language Programming.
4	Study of 8086 microprocessors with interfacing of peripheral devices.
5	Realize techniques for faster execution of instructions and improve speed of operation and performance of microprocessors.
6	Write and debug programs in TASM/MASM/hardware kits
Lab Outcomes:On successful completion, of course, learner/student will be able to:	
1	Demonstrate various components and peripheral of computer system
2	Analyze and design combinational circuits
3	Build a program on a microprocessor using arithmetic & logical instruction set of 8086.
4	Develop the assembly level programming using 8086 loop instruction set
5	Write programs based on string and procedure for 8086 microprocessors.
6	Design interfacing of peripheral devices with 8086 microprocessors.

4)Suggested Experiments: (minimum number of experiments to be completed can be specified)	
Sr. No.	**Name of the Experiment**
1	PC Assembly:Study of PC Motherboard Technology (South Bridge and North Bridge), Internal Components and Connections used in computer system.
2	Implementation of combinational circuits:1. Verify the truth table of various logic gates (basic and universal gates) 2. Realize Half adder and Full adder 3. Implementation of MUX and DeMUX 4.Implementation of Encoder and Decoder
3	Arithmetic and logical operations in 8086 Assembly language programming: 1. Program for 16 bit BCD addition 2. Program to evaluate given logical expressions. 3. Convert two digit Packed BCD to Unpacked BCD. 4. Multiply 8 bit and 16 bit numbers
4	Loop operations in 8086 Assembly language programming : 1. Program to move set of numbers from one memory block to another. 2. Program to count number of 1's and 0's in a given 8 bit number 3. Program to add 10 numbers using a loop.
5	3. Program to find even and odd numbers from a given list 4. Program to search for a given number
6	String & Procedure in 8086 Assembly language programming: 1. Check whether a given string is a palindrome or not. 2. Compute the factorial of a positive integer 'n' using procedure.
7	Check whether a given string is a palindrome or not.
8	2. Compute the factorial of a positive integer 'n' using procedure.
9	Generate the first 'n' Fibonacci numbers.
10	ALP to sort a given set of 16 bit unsigned integers into ascending order using bubble sort algorithm
11	ALP to find smallest no from the given array

12	Case study on Interfacing with 8086 microprocessor 1. Interfacing Seven Segment Display 2. Interfacing keyboard matrix

5)Text books:	
1	Scott Mueller, ”Upgrading and repairing PCs”, Pearson,
2	R. P. Jain, “Modern Digital Electronics”, Tata McGraw Hill.
3	John Uffenbeck, “8086/8088 family: Design Programming and Interfacing:”Pearson Education
4	M. Morris Mano, “Digital Logic and computer Design”, PHI
	K Bhurchandi, “Advanced Microprocessors & Peripherals”, Tata McGraw-Hill Education

6) Term Work:	
1	**Term Work shall consist of at least 10 Practical based on the above list, but not limited to. 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)**
7) Continuous assessment exam:	
1.	Timely Submission of Experiments weekwise
2.	Explanation/concept:
3.	Algorithm/implementation:
4.	Analysis
5.	Documentation/Performance:

Lab Code	Lab Name	Credit
ITL303	UNIX Lab	**1**

1)Prerequisite: The Lab experiments aims:	
2)Lab Objectives:	
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.
3)Lab Outcomes: On successful completion, of course, learner/student will be able to:	
1	Understand the architecture and functioning of Unix
2	Identify the Unix general purpose commands
3	Apply Unix commands for system administrative tasks such as file system management and user management
4	Execute Unix commands for system administrative tasks such as process management and memory management
5	Implement basic shell scripts for different applications
6	Implement advanced scripts using awk & perl languages and grep, sed, etc. commandsfor performing various tasks

6) Term Work:	
1	Term Work shall consist of at least 10 Practical based on the above list, but not limited to. 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)

7) Continuous assessment exam:	
1.	Timely Submission of Experiments weekwise
2.	Explanation/concept:
3.	Algorithm/implementation:
4.	Analysis
5.	Documentation/Performance:

Course Code: ITC404	**Course Title :Python Lab (SBL)**	**Credit**
Currently same	**Python Lab (SBL)**	02
1)Prerequisite: C, Java programming		
2)Lab Objectives: The course aims:		
1	Basics of python including data types, operator, conditional statements, looping statements, input and output functions in Python.	
2	List, tuple, set, dictionary, string, array and functions.	
3	Object Oriented Programming concepts in python.	
4	Concepts of modules, packages, multithreading and exception handling.	
5	File handling, GUI & database programming.	
6	Data visualization using Matplotlib, Data analysis using Pandas and Web programming using Flask.	
3)Lab Outcomes: On successful completion, of course, learner/student will be able to:		
1	Understand the structure, syntax, and semantics of the Python language.	
2	Interpret advanced data types and functions in python.	
3	Illustrate the concepts of object-oriented programming as used in Python.	
4	Create Python applications using modules, packages, multithreading and exception handling.	
5	Gain proficiency in writing File Handling programs ,also create GUI applications and evaluate database operations in python.	
6	Design and Develop cost-effective robust applications using the latest Python trends and technologies.	

4) syllabus

Module		Content	Hrs	
I	**Basics of Python**	Introduction, Features, Python building blocks – Identifiers, Keywords, Indention, Variables and Comments, Basic data types (Numeric, Boolean, Compound) Operators: Arithmetic, comparison, relational, assignment, logical, bitwise, membership, identity operators, operator precedence Control flow statements: Conditional statements (if, if…else, nested if) Looping in Python (while loop, for loop, nested loops) Loop manipulation using continue, pass, break. Input/output Functions, Decorators, Iterators and Generators.	**08**	LO 1
II	**Advanced data types & Functions**	Lists: a) Defining lists, accessing values in list, deleting values in list, updating lists b) Basic list operations c) Built-in list functions Tuples: a) Accessing values in Tuples, deleting values in Tuples, and updating Tuples b) Basic Tuple operations c) Built-in Tuple functions Dictionaries: a) Accessing values in Dictionary, deleting values in Dictionary, and updating Dictionary b) Basic Dictionary operations c) Built-in Dictionary functions Sets: a) Accessing values in Set, deleting values in Set, updating Sets b) Basic Set operations, c) Built-in Set functions Strings: a) String initialization, Indexing, Slicing, Concatenation, Membership & Immutability b) Built-in String functions Arrays: a) Working with Single dimensional Arrays: Creating, importing, Indexing, Slicing, copying and processing array arrays. b) Working with Multi-dimensional Arrays using Numpy: Mathematical operations, Matrix operations, aggregate and other Built-in functions Functions: a) Built-in functions in python b) Defining function, calling function, returning values, passing parameters c) Nested and Recursive functions d) Anonymous Functions (Lambda, Map, Reduce, Filter)	**09**	LO 1 LO 2
III	**Object Oriented Progr**	Overview of Object-oriented programming, Creating Classes and Objects, Self-Variable, Constructors, Inner class, Static method, Namespaces. Inheritance: Types of Inheritance (Single, Multiple, Multi-level, Hierarchical), Super() method, Constructors in inheritance, operator overloading, Method overloading, Method overriding, Abstract class, Abstract method, Interfaces in Python.	**08**	LO 1 LO 3

	amming			
IV	**Exploring concept of modules, packages, multithreading and exception handling**	Modules: Writing modules, importing objects from modules, Python built-in modules (e.g. Numeric and Mathematical module, Functional Programming module, Regular Expression module), Namespace and Scoping. Packages: creating user defined packages and importing packages. Multi-threading: process vs thread, use of threads, types of threads, creating threads in python, thread synchronization, deadlock of threads. Exception handling: Compile time errors, Runtime errors, exceptions, types of exception, try statement, except block, raise statement, Assert statement, User-Defined Exceptions.	**06**	LO 1 LO 4
V	**File handling, GUI & database programming**	File Handling: Opening file in different modes, closing a file, writing to a file, accessing file contents using standard library functions, reading from a file – read (), readline (), readlines (), Renaming and Deleting a file, File Exceptions, Pickle in Python. Graphical user interface (GUI): different GUI tools in python (Tkinter, PyQt, Kivy etc.), Working with containers, Canvas, Frame, Widgets (Button, Label, Text, Scrollbar, Check button, Radio button, Entry, Spinbox, Message etc.) Connecting GUI with databases to perform CRUD operations. (on supported databases like SQLite, MySQL, Oracle, PostgreSQL etc.).	**09**	LO 1 LO 5
VI	**Data visualization, analysis and web programming using python**	Visualization using Matplotlib: Matplotlib with Numpy, working with plots (line plot, bar graph, histogram, scatter plot, area plot, pie chart etc.), working with multiple figures. Data manipulation and analysis using Pandas: Introduction to Pandas, importing data into Python, series, data frames, indexing data frames, basic operations with data frame, filtering, combining and merging data frames, Removing Duplicates. SciPy: Linear algebra functions using Numpy and Scipy. Web programming: Introduction to Flask, Creating a Basic Flask Application, Build a Simple REST API using Flask	**10**	LO 1 LO 6

		Total	50	

5) Textbooks:	
1	Dr. R. Nageswara Rao,” Core Python Programming” , Dreamtech Press, Wiley Publication
2	M. T. Savaliya , R. K. Maurya, “Programming through Python”, StarEdu Solutions.
3	E Balagurusamy, “Introduction to computing and problem-solving using python”, McGraw Hill Publication.
6) Reference Books:	
1	Zed A. Shaw, “Learn Python 3 the Hard Way”, Zed Shaw's Hard Way Series.
2	Martin C. Brown,” Python: The Complete Reference”, McGraw-Hill Publication.
3	Paul Barry,” Head First Python”, 2nd Edition, O'Reilly Media, Inc.

7) Internal Assessment:
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)

Sr.r	Rubrics	Marks
1.	***Certificate course for 4 weeks or more:- NPTEL/Coursera/Udemy/any MOOC**	**10 marks**
2.	**Extra Experiments/ Virtual Lab /GATE Based Assignment test/Tutorials etc**	**10 marks**
3.	**Continuous Assesment**	**20 Marks**

*****For sr.no.1, the date of the certification exam should be within the term and in case a student is unable to complete the certification , the grading has to be done accordingly.

9)Rubrics for slow learners:-

1.) Case study, Presentation, group discussion, technical debate on recent trends in the said course (10 marks)

2. Project based Learning and evaluation / Extra assignment / Question paper solution (10 marks)

3) Multiple Choice Questions (Quiz) (5marks)

4) Literature review of papers/journals (5 marks)

5) Library related work (5 marks)

10)Rubrics for Indirect Assessment :-

1. Mock Viva/Practical

2. Skill Enhancement Lecture

3. Extra Assignments/lab/lecture

Course Code: ITM401	**Course Title : Mini Project – 1 B for Python based automation projects**	**Credit**
Currently same	**Mini Project – 1 B for Python based automation projects**	
1)Prerequisite:		
2)Course Objectives: The course aims:		
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.	
3)Course Outcomes: On successful completion, of course, learner/student 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.

4) 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.

5)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

6) 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.

7) 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.

8) 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