B. TECH.

COMPUTER SCIENCE AND ENGINEERING

(ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

THIRD SEMESTER

STUDY AND EVALUATION SCHEME

B.TECH. COMPUTER SCIENCE AND ENGINEERING
(ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

II Year: III SEMESTER

S. No.	Subject Code	Subject Name	L	T	P	Sessional Assessment			SEE	Subject Total	Credit
						MST	TA	Total			
THEORY SUBJECT											
1	TAS301	Engineering Mathematics - III	3	1	0	15	15	30	70	100	4
2	TCS301	Data Structures and Algorithms	3	1	0	15	15	30	70	100	4
3	TCS302	Python Programming	3	0	0	15	15	30	70	100	3
4	TEC301	Digital Electronics	3	0	0	15	15	30	70	100	3
5	THM301	Universal Human Value and Professional Ethics	2	0	0	15	15	30	70	100	2
6	TGP301	General Proficiency	-	-	-	-	-	50	0	50	0
PRACTICAL/DESIGN/DRAWING											
7	TCS351	Data Structures and Algorithms Lab	0	0	2	15	15	30	70	100	1
8	TCS352	Python Programming Lab	0	0	2	15	15	30	70	100	1
9	TCS353	Web Designing Workshop	0	0	2	15	15	30	70	100	1
10	TEC351	Digital Electronics Lab	0	0	2	15	15	30	70	100	1
		Total	**14**	**2**	**8**	**135**	**135**	**320**	**630**	**950**	**20**

L- Lecture
T -Tutorial
P-Practical
MST- Mid Semester Test
TA-Teacher's Assessment
SEE- Semester End Examination

MATHEMATICS-III
(TAS301)

COURSE OUTCOMES (COs):	
CO 1:	Compute Laplace Transform and Inverse Laplace Transforms.
CO 2:	Gain Knowledge of Moments, Curve Fitting, Correlation and Regression, Probability Distribution and Chi Square Test.
CO 3:	Implement Solution of Transcendental and Polynomial Equations, Finite Differences and Interpolation.
CO 4:	Compute Solution of System of Linear Equations, Numerical Differentiation and Integration as well as Solution of Ordinary Differential Equations.
CO 5:	Apply Fourier Transforms and Z-Transforms.
Unit	**Topic**
I	**Laplace Transform:** Laplace transform, Existence theorem, Laplace transforms of derivatives and integrals, Initial and final value theorems, Unit step function, Dirac-delta function, Laplace transform of periodic function, Inverse Laplace transform, Convolution theorem, Application to solve simple linear and simultaneous differential equations.
II	**Statistical Techniques:** Moments, Moment generating functions, Skewness, Kurtosis, Curve fitting, Method of least squares, Fitting of straight lines, Polynomials, Exponential curves, Correlation, Linear, non-linear and multiple regression analysis, Binomial, Poisson and Normal distributions, Tests of significations: Chi-square test, t-test.
III	**Numerical Techniques - I:** Zeroes of transcendental and polynomial equations using Bisection method, Regula- falsi method and Newton-Raphson method, Rate of convergence of above methods. Interpolation: Finite differences, Newton's forward and backward interpolation, Lagrange's and Newton's divided difference formula for unequal intervals.
IV	**Numerical Techniques - I:** Zeroes of transcendental and polynomial equations using Bisection method, Regula- Falsi method and Newton-Raphson method, Rate of convergence of above methods. Interpolation: Finite differences, Newton's forward and backward interpolation, Lagrange's and Newton's divided difference formula for unequal intervals.
V	**Integral Transforms:** Fourier integral, Complex Fourier transform, Inverse Transforms, Convolution Theorems, Fourier sine and cosine transform, Applications of Fourier transform to simple one-dimensional heat transfer equations, wave equations and Laplace equations, Z- transform and its application to solve difference equations.

Text Book (s):

1. 1. H.K. Das, Advanced Engineering Mathematics, S. Chand and Company Pvt. Ltd.,2022
2. Peter V. O'Neil, Advance Engineering Mathematics Thomson (Cengage) Learning. 2007.
3. Jain, Iyenger and Jain, Numerical Methods for Scientific and Engineering Computation, New Age International, New Delhi.,2020
4. J N Kapur, Mathematical Statistics, S. Chand and Company Pvt. Ltd.2014
5. BS Grewal, Higher Engineering Mathematics, Khanna Publishers.,1965

DATA STRUCTURES AND ALGORITHMS
(TCS301)

COURSE OUTCOMES (COs):	
CO 1:	Identifying and understanding various data structures that can be implemented using different programming languages.
CO 2:	Selecting appropriate data structures that effectively represent the information within a given problem and evaluating the efficiency trade-offs, including runtime
CO 3:	Designing, implementing, testing, and debugging programs utilizing a range of data structures, such as stacks, queues, binary trees, search trees, and graphs.
CO 4:	Understanding the concept of recursion, application of recursion and its implementation and removal of recursion
CO 5:	Applying efficient data structures, including linked lists, stacks, and queues, to address and solve specific problems.
Unit	**Topic**
I	**Introduction and Elementary Data Structures:** **Introduction**: Introduction to Data Structures and data types, Efficient use of memory, Recursion, time and space complexity of algorithms, Big O Notation and theta notations. **Elementary Data Structures:** Stacks, queues, Infix, Postfix and Prefix conversions, evaluations of expressions, Queues, Priority queues, double ended queue, implementation of stacks and queues, Operations on data structures.
II	**Linked lists:** Introduction to linked list, linked stacks and queues, Array Implementation and Dynamic Implementation of Singly Linked Lists, Doubly Linked List, Circularly Linked List; polynomial addition, sparse matrices Applications of Stacks, Queues and Linked lists, Garbage collection.
III	**Trees:** Basic terminology, binary trees, binary tree traversal, Array and Linked Representation of Binary trees, application of trees, Threaded Trees, Binary Search Tree, AVL tree, B-tree.
IV	**Graph Theory:** Basic terminology, Sequential and linked Representations of Graphs, Graph Traversals, Dijkstra's algorithm for shortest path, Prim's and Kruskal's Algorithm for Minimal Spanning tree.
V	**Sorting and Searching Algorithm:** **Searching**: Linear search, binary search and hash search. **Sorting:** Insertion sort, selection sort, bubble sort, quick sort, merge sort, heap sort and Bucket sort.

Text Book (s):

1. "Fundamentals of Data Structures", Illustrated Edition by Ellis Horowitz, Sartaj Sahni, Computer Science Press.
2. "Data Structures", RS Salaria, Khanna Publishing House
3. How to Solve it by Computer", 2nd Impression by R.G. Dromey, Pearson Education
4. Expert Data Structures with C/ 3rd Edition, R.B. Patel, Khanna Book Publishing, 2020.
5. An introduction to data structures with applications, By J.P. Trembley and P.G. Sorensen, TMH, 2004.

PYTHON PROGRAMMING
(TCS302)

COURSE OUTCOMES (COs):	
CO 1:	Understanding Fundamentals of Python Programming
CO 2:	Understand and implement Control Structures.
CO 3:	Learn and implement Strings and Functions in Python.
CO 4:	Understand and implement advance functions like iteration and recursion.
CO 5:	Implement Object Oriented Programming concepts in Python
Unit	**Topic**
I	**Introduction:** The Programming Cycle for Python, Python IDE, Interacting with Python Programs, Elements of Python, Type Conversion. Basics: Expressions, Assignment Statement, Arithmetic Operators, Operator Precedence, Boolean Expression.
II	**Conditionals and Loops:** Conditional statement in Python: if-else statement, its working and execution, Nested-if statement and Else-if statement in Python, Expression Evaluation and Float Representation, Loops: Purpose and working of loops, while loop including its working, For Loop, Nested Loops, Break and Continue.
III	**Strings and Functions:** Strings: Length of the string, Concatenation and Repeat operations, Indexing and Slicing of Strings. Python Data Structure: Tuples, Unpacking Sequences, Lists, Mutable Sequences, List Comprehension, Sets, Dictionaries, Functions: Parts of a Function, Execution of a Function, Keyword and Default Arguments, Scope Rules, Higher Order Functions: Treat functions as first-class Objects, Lambda Expressions.
IV	**Classes and Files:** Generate prime numbers with the help of Sieve of Eratosthenes algorithm, File I/O: File input and output operations in Python Programming Exceptions and Assertions Modules: Introduction, Importing Modules, Abstract Data Types: Abstract data types and ADT interface in Python Programming, Classes: Definition and operations in the classes, Special Methods (such as _init_, _str_, comparison methods and Arithmetic methods etc.), Class Example, Inheritance, Inheritance and OOP.
V	**Iterators and Recursion:** Recursive Fibonacci, Tower of Hanoi, Search: Simple Search, Binary Search, Estimating Search Time in Simple Search and Binary Search, Sorting and Merging: Selection Sort, Merge List, Merge Sort, Higher Order Sort.

Text Book (s):

1. Allen B. Downey, ``Think Python: How to Think Like a Computer Scientist, 2nd edition, Updated for Python 3, Shroff/O,,Reilly Publishers, 2016, (http://greenteapress.com/wp/thinkpython/)
2. Guido van Rossum and Fred L. Drake Jr, An Introduction to Python – Revised and updated for Python 3.2, Network Theory Ltd., 2011.
3. John V Guttag, ―Introduction to Computation and Programming Using Python, Revised and expanded Edition, MIT Press, 2013.
4. Python Training Guide –Mercury Learning and Information USA, BPB Publications.
5. Lutz, Mark, Programming Python: Powerful Object-Oriented Programming, O Rielly

DIGITAL ELECTRONICS
(TEC301)

COURSE OUTCOMES (COs):	
CO 1:	Identify various number systems, binary codes and formulate digital function using Boolean algebra.
CO 2:	Design and implement combinational logic circuits.
CO 3:	Design and implement sequential logic circuits.
CO 4:	Compare the operation of various analog to digital and digital to analog conversion circuits.
CO 5:	Explain the basic concepts of programmable logic devices and VHDL.
Unit	**Topic**
I	Number Systems and Codes: Binary, Octal and hexadecimal conversions-ASCII code, Excess -3 code, gray code, BCD, Error detection codes-Parity method. Signed numbers- representation, addition and subtraction, Fixed point and floating-point representation. Logic gates, Universal gates, TTL and CMOS logic families-Internal diagram of TTL NAND gate and CMOS NOR gate. Comparison of CMOS and TTL performance.
II	Boolean Laws and theorems, Sum of Products method, Product of Sum method – K map representation and simplification (up to four variables) - Pairs, Quads, Octets, don't care conditions. Combinational circuits: Adders -Full adder and half adder, Subtractors- half subtractor and full subtractor, 4 - bit parallel binary adder / subtractor, Carry Look ahead adders.
III	Comparators, Parity generators and checkers, Encoders, Decoders, BCD to seven segment decoder, Code converters, Multiplexers, Demultiplexers, Architecture of Arithmetic Logic Units (Block schematic only).
IV	Flip-Flops, SR, JK, D and T flip-flops, JK Master Slave Flip-flop, Preset and clear inputs, Conversion of flip-flops. Registers -SISO, SIPO, PISO, PIPO. Up/Down Counters: Asynchronous Counters – Modulus of a counter – Mod-N counters Ring counter, Johnson Counter Synchronous counters, Design of Synchronous counters.
V	State Machines: State transition diagram, Moore and Mealy Machines Digital to Analog converter –Specifications, Weighted resistor type, R-2R Ladder type.Analog to Digital Converter – Specifications, Flash type, Successive approximation type. Programmable Logic Devices - PAL, PLA, FPGA (Introduction and basic concepts only) Introduction to Verilog, Implementation of AND, OR, half adder and full adder.

Text Book (s):

1. Floyd T.L, Digital Fundamentals, 10/e, Pearson Education, 2011.
2. C.H.Roth and L.L.Kimney Fundamentals of Logic Design, 7/e, Cengage Learning,
3. Mano M.M, Logic and Computer Design Fundamentals, 4/e, Pearson Education.
4. A Anand Kumar, Fundamental of Digital Electronics, Prentice Hall
5. Roy Chaudari, Linear Integrated Circuits, New Age International Publications
6. S. Saliva Hanan, Digital Circuits and Design, Oxford University Press

UNIVERSAL HUMAN VALUE AND PROFESSIONAL ETHICS
(THM301)

COURSE OUTCOME (COs):	
CO 1:	Identify and analyze an ethical issue in the subject matter under investigation or in a relevant field
CO 2:	Identify the multiple ethical interests at stake in a real-world situation or practice
CO 3:	Articulate what makes a particular course of action ethically defensible
CO 4:	Assess their own ethical values and the social context of problems
CO 5:	Identify ethical concerns in research and intellectual contexts, including academic integrity, use and citation of sources, the objective presentation of data, and the treatment of human.

Unit	Topic
I	**HUMAN VALUES:** Morals, Values and Ethics-Integrity-Work Ethic-Service learning – Civic Virtue – Respect for others –Living Peacefully –Caring –Sharing –Honesty -Courage-Cooperation– Commitment – Empathy –Self Confidence Character –Spirituality-Case Study.
II	**ENGINEERING ETHICS:** Senses of 'Engineering Ethics-Variety of moral issued –Types of inquiry –Moral dilemmas – Moral autonomy –Kohlberg's theory-Gilligan's Theory-Consensus and controversy –Models of professional roles-Theories about right action-Self-interest -Customs and religion –Uses of Ethical theories –Valuing time –Co-operation –Commitment-Case Study.
III	**ENGINEERING AS SOCIAL EXPERIMENTATION:** Engineering as Social Experimentation –Framing the problem –Determining the facts – Codes of Ethics –Clarifying Concepts –Application issues –Common Ground - General Principles –Utilitarian thinking respect for persons-Case study.
IV	**ENGINEERS RESPONSIBILITY FOR SAFETY AND RISK:** Safety and risk –Assessment of safety and risk –Risk benefit analysis and reducing risk- Safety and the Engineer-Designing for the safety-Intellectual Property Rights (IPR).
V	**GLOBAL ISSUES:** Globalization –Cross culture issues-Environmental Ethics – Computer Ethics –Computers as the instrument of Unethical behavior – Computers as the object of Unethical acts – Autonomous Computers-Computer codes of Ethics –Weapons Development -Ethics and Research –Analyzing Ethical Problems in research- Case Study

Text Book (s):

1. M.Govindarajan, S.Natarajananad, V.S.SenthilKumar "Engineering Ethics includes Human Values" -PHI Learning Pvt. Ltd-2009
2. Harris, Pritchard and Rabins "Engineering Ethics", CENGAGE Learning, India
3. Mike W. Martin and Roland Schinzinger "Ethics in Engineering" TMH.
4. Prof. A. R. Aryasri, Dharanikota Suyodhana "Professional Ethics and Morals", Maruthi Publications.
5. A. Alavudeen, R. Kalil Rahman and M.Jayakumaran "Professional Ethics and Human Values" -Laxmi Publications.

DATA STRUCTURES AND ALGORITHMS LAB
(TCS351)

LIST OF THE EXPERIMENTS:

1. Implementing Sorting Techniques: Bubble Sort, Insertion Sort, Selection Sort, Shell , Sort, Radix Sort, Quick sort.

2. Implementing Searching and Hashing Techniques: Linear search, Binary search, Methods for Hashing: Modulo Division, Digit Extraction, Fold shift, Fold Boundary, Linear Probe for Collision Resolution. Direct and Subtraction hashing.

3. Implementing Stacks: Array implementation, Linked List implementation, Evaluation of postfix expression and balancing of parenthesis, Conversion of infix notation to postfix notation.

4. Implementing Queue: Linked List implementation of ordinary queue, Array implementation of circular queue, Linked List implementation of priority queue, Double ended queue.

5. Implementing Linked List: Singly Linked Lists, Circular Linked List, Doubly Linked Lists : Insert, Display, Delete, Search, Count, Reverse(SLL), Polynomial , Addition , Comparative study of arrays and linked list.

6. Implementing Trees: Binary search tree: Create, Recursive traversal: preorder, post order, in order, Search Largest, Node, Smallest Node, Count number of nodes, Heap: Min Heap, Max Heap: reheap Up, reheap Down, Delete, Expression Tree, Heapsort.

7. Implementing Graphs: Represent a graph using the Adjacency Matrix, BFS, Find the minimum spanning tree (using any method Kruskal's Algorithm or Prim's Algorithm) Self Learning Topics: Shortest Path Algorithm.

Note: The instructor may add/ modify the list as per his/her wisdom for better hands on the students.

PYTHON PROGRAMMING LAB
(TCS352)

LIST OF THE EXPERIMENTS:

1. To read and write simple Python programs.
2. To develop Python programs with conditionals and loops.
3. To define Python functions and to use Python data structures –- lists, tuples, dictionaries
4. To do input/output with files in Python
5. Write a Python Program to perform Linear Search
6. Write a Python Program to perform Binary Search
7. Write a Python Program to perform selection sort
8. Write a Python Program to perform insertion sort.

Note: The instructor may add/ modify the list as per his/her wisdom for better hands on the students.

DIGITAL ELECTRONICS LAB
(TEC351)

LIST OF THE EXPERIMENTS:

1. Truth table verification of different logic gates- AND, OR, NOT, NAND, NOR, X-OR, and X-NOR
2. Realization of basic Gates using universal Gates.
3. Design of half and full adder circuits.
4. Design of half and full subtractor circuits.
5. Design of code converters circuits-Binary to Gary code, and Gray to Binary code.
6. Design of magnitude comparator circuits.
7. Study of multiplexer (MUX) and demultiplexer (DEMUX) circuits.
8. Design of shift register.
9.. Study of different types of Flip Flops (FFs)-SR-FF, JK-FF, D-FF, and T-FF.
10. Conversion from one Flip Flop (FF) to others.

Note: The instructor may add/ modify the list as per his/her wisdom for better hands on the students.

WEB DESIGNING WORKSHOP
(TCS353)

LIST OF THE EXPERIMENTS:

1. Write a program in HTML to display different styles of heading text.
2. Write a program to display the processes to be followed for a patient when he enters for a complete check-up. Use ordered lists and unordered lists.
3. Write a program to display a traditional Newspaper with the use of table tags.
4. With the help of "IMAGE" tags write a program to display the image along with some contents.
5. Use "Anchor" tag to write a program for displaying various Menus.
6. Use mapping technique, to map a particular part of image and move the control corresponding to that area. For e.g., in an image, if there are bat, ball, stump etc. When you click stump control should move to a file call St.htm.
7. Create frames that have details about various cities.
8. Create a form to display the kinds of food available in a Restaurant. (Use checkboxes wherever necessary)
9. Write a program to "reload" a page automatically once in 5 seconds.
10. Write a program using CSS to set the background colour, font, and paragraph

Note: The instructor may add/ modify the list as per his/her wisdom for better hands on the students.

B. TECH.

COMPUTER SCIENCE AND ENGINEERING

(ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

FOURTH SEMESTER

STUDY AND EVALUATION SCHEME

B.TECH. COMPUTER SCIENCE AND ENGINEERING
(ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

II Year: IV Semester

S. No.	Subject Code	Subject name	L	T	P	Sessional Assessment			SEE	Subject Total	Credit
						MST	TA	Total			
THEORY SUBJECT											
1	TCS401	Design and Analysis of Algorithms	3	1	0	15	15	30	70	100	4
2	TCS401	Object Oriented Programming with Java	3	0	0	15	15	30	70	100	3
3	TCS403	Computer Organisation and Architecture	3	0	0	15	15	30	70	100	3
4	TCS404	Database Management System	3	1	0	15	15	30	70	100	4
5	THM401	Technical Communication	2	0	0	15	15	30	70	100	2
6	TGP401	General Proficiency	-	-	-	-	-	50	0	50	0
PRACTICAL/DESIGN/DRAWING											
7	TCS451	Design and Analysis of Algorithm Lab	0	0	2	15	15	30	70	100	1
8	TCS452	Object Oriented Programming with Java Lab	0	0	2	15	15	30	70	100	1
9	TCS453	Computer Organisation and Architecture Lab	0	0	2	15	15	30	70	100	1
10	TCS454	Database Management System Lab	0	0	2	15	15	30	70	100	1
		Total	14	2	8	135	135	320	630	950	20

*** The Mini Project or internship (4 weeks) will be done during summer break after 4th Semester and will be assessed during V semester.**

L- Lecture
T -Tutorial
P-Practical
MST- Mid Semester Test
TA-Teacher's Assessment
SEE- Semester End Examination

DESIGN AND ANALYSIS OF ALGORITHMS
(TCS401)

COURSE OUTCOMES (COs):	
CO 1:	Apply the most appropriate data structure to design an algorithm for solving a specified problem.
CO 2:	Evaluates various algorithms in terms of their time and space complexity.
CO 3:	Develop algorithms to effectively address a wide-range of computational challenges.
CO 4:	Understand different complexity classes.
CO 5:	Understand basic techniques for designing algorithms, including the techniques of recursion, divide-and-conquer, and greedy

Unit	Topic
I	**Basic Concepts of Algorithms:** Notion of Algorithm, Fundamentals of Algorithmic Solving, Important problem types, Fundamentals of the Analysis Framework, Asymptotic Notations and Basic Efficiency Classes, Mathematical analysis of non-recursive algorithms. Mathematical analysis of recursive algorithm: recurrence relations, solution of recurrence relations using substitution method.
II	**Brute Force, Divide and Conquer Strategy:** Selection sort, Bubble sort, Sequential searching (Linear Search), Brute force string matching, General method, Merge sort, Quick Sort, Binary Search.
III	**Greedy Approach and Dynamic Programming** Fractional Knapsack problem, Minimum cost spanning tree: Prim's and Kruskal's algorithm, Single source shortest path problem, Principle of optimality, all pair shortest path problem, 0/1 Knapsack problem, Traveling salesperson problem.
IV	**backtracking and Branch and Bound:** General method backtracking, N-Queen problem, 0/1 Knapsack problem, General method of branch and bound, 0/1 Knapsack problem, Traveling sales person problem
V	**String Matching Algorithms:** Naïve string-matching algorithm, Rabin-Karp algorithm, Knuth-Morris-Pratt algorithm. Introduction of NP-completeness, Randomized algorithms and Approximation Algorithms

Text Book (s):

1. Introduction to Algorithms, Thomas H Cormen, Charles E Lieserson, Ronald L Rivest and Clifford Stein, MIT Press/McGraw-Hill.
2. Fundamentals of Algorithms – E. Horowitz et al.
3. Design and Analysis of Algorithms, S. Sridhar, Oxford
4. Design and Analysis of Algorithms, Gajendra Sharma, Khanna Book Publishing 2018.
5. Algorithm Design, Jon Kelinberg and Eva Tardos, 1st Edition, Pearson Education 2014.
6. Data Structures and Program Design in C By Robert L. Kruse, C.L. Tondo, Bruce Leung, Pearson Education. 2007.

OBJECT ORIENTED PROGRAMMIING WITH JAVA
(TCS402)

COURSE OUTCOMES (COs):

CO 1:	Develop the object-oriented programming concepts using Java.
CO 2:	Implement exception handling, file handling, and multi-threading in Java.
CO 3:	Apply new java features to build java programs.
CO 4:	Analyse java programs with Collection Framework.
CO 5:	Test Web Services with Spring Boot using Spring Framework Concepts.

Unit	Topic
I	**Introduction:** Why Java, History of Java, JVM, JRE, Java Environment, Java Source File Structure, and Compilation. Fundamental. **Programming Structures in Java:** Defining Classes in Java, Constructors, Methods, Access Specifies, Static Members, Final Members, Comments, Data types, Variables, Operators, Control Flow, Arrays and String. **Object Oriented Programming:** Class, Object, Inheritance Super Class, Sub Class, Overriding, Overloading, Encapsulation, Polymorphism, Abstraction, Interfaces, and Abstract Class. **Packages:** Defining Package, CLASSPATH Setting for Packages, Making JAR Files for Library Packages, Import and Static Import Naming Convention for Packages.
II	**Exception Handling:** The Idea behind Exception, Exceptions and Errors, Types of Exception, Control Flow in Exceptions, JVM Reaction to Exceptions, Use of try, catch, finally, throw, throws in Exception Handling, In-built and User Defined Exceptions, Checked and Un-Checked Exceptions. **Input /Output Basics:** Byte Streams and Character Streams, Reading and Writing File in Java. **Multithreading:** Thread, Thread Life Cycle, Creating Threads, Thread Priorities, Synchronizing Threads, Inter-thread Communication.
III	**Java New Features:** Functional Interfaces, Lambda Expression, Method References, Stream API, Default Methods, Static Method, Base64 Encode and Decode, ForEach Method, Try-with resources, Type Annotations, Repeating Annotations, Java Module System, Diamond Syntax with Inner Anonymous Class, Local Variable Type Inference, Switch Expressions, Yield Keyword, Text Blocks, Records, Sealed Classes
IV	**Java Collections Framework:** Collection in Java, Collection Framework in Java, Hierarchy of Collection Framework, Iterator Interface, Collection Interface, List Interface, Array List, LinkedList, Vector, Stack, Queue Interface, Set Interface, HashSet, Linked HashSet, Sorted Set Interface, Tree Set, Map Interface, HashMap Class, Linked HashMap Class, Tree Map Class, Hash table Class, Sorting, Comparable Interface, Comparator Interface, Properties Class in Java.
V	**Spring Framework:** Spring Core Basics-Spring Dependency Injection concepts, Spring Inversion of Control, AOP, Bean Scopes- Singleton, Prototype, Request, Session, Application, Web Socket, Auto wiring, Annotations, Life Cycle Call backs, Bean Configuration styles. **Spring Boot:** Spring Boot Build Systems, Spring Boot Code Structure, Spring Boot Runners, Logger, BUILDING RESTFUL WEB SERVICES, Rest Controller, Request Mapping, Request Body, Path Variable, Request Parameter, GET, POST, PUT, DELETE APIs, Build Web Applications

Text Book (s):

1. Herbert Schildt, "Java The complete reference", McGraw Hill Education
2. Craig Walls, “Spring Boot in Action” Manning Publication
3. Steven Holzner, “Java Black Book”, Dreamtech.

COMPUTER ORGANIZATION AND ARCHITECTURE
(TCS403)

COURSE OUTCOMES (COs):	
CO 1:	Study of the basic structure and operation of a digital computer system.
CO 2:	Analysis of the design of arithmetic and logic unit and understanding of the fixed point and floating-point arithmetic operations.
CO 3:	Implementation of control unit techniques and the concept of Pipelining
CO 4:	Understanding the hierarchical memory system, cache memories and virtual memory
CO 5:	Understanding the different ways of communicating with I/O devices and standard I/O interfaces
Unit	**Topic**
I	Introduction: Functional units of digital system and their interconnections, buses, bus architecture, types of buses and bus arbitration. Method of bus, register and memory transfer. Processor organization, general registers organization, stack organization and addressing modes.
II	Arithmetic and logic unit: Look ahead carries adders. Multiplication: Signed operand multiplication, Booth's algorithm and array multiplier. Division and logic operations. Floating point arithmetic operation, Arithmetic and logic unit design. IEEE Standard for Floating Point Numbers
III	Control Unit: Instruction types, formats, instruction cycles and sub cycles (fetch and execute etc), micro-operations, execution of a complete instruction. Program Control, Reduced Instruction Set Computer, Pipelining. Hardwire and micro programmed control: micro programme sequencing, concept of horizontal and vertical microprogramming.
IV	Memory: Basic concept and hierarchy, semiconductor RAM memories, 2D and 2 1/2D memory organization. ROM memories. Cache memories: concept and design issues and performance, address mapping and replacement Auxiliary memories: magnetic disk, magnetic tape and optical disks Virtual memory: concept implementation.
V	Input / Output: Peripheral devices, I/O interface, I/O ports, Interrupts: interrupt hardware, types of interrupts and exceptions. Modes of Data Transfer: Programmed I/O, interrupt initiated I/O and Direct Memory Access., I/O channels and processors. Serial Communication: Synchronous and asynchronous communication, standard communication interfaces.

Text Book (s):

1. M. Morris Mano "Computer System Architecture" Third Edition, Pearson Education India, Pvt. Ltd, reprint 2007
2. David A. Patterson and John. L. Hennessey "Computer Organization and Design, The Hardware/Software Interface", 3rd Edition, Morgan Kaufmann Publishers.
3. Carl Hamacher, Zvonko Vranesic, Safwat Zaky Computer Organization, McGraw-Hill, Fifth Edition, Reprint 2012
4. William Stallings, Computer Organization and Architecture-Designing for Performance, Pearson Education, Seventh edition, 2006.
5. Behrooz Parahami, "Computer Architecture", Oxford University Press, Eighth Impression, 2011.

DATABASE MANAGEMENT SYSTEM
(TCS404)

COURSE OUTCOMES (COs):	
CO 1:	Apply knowledge of database for real life applications.
CO 2:	Apply query processing techniques to automate the real time problems of databases.
CO 3:	Identify and solve the redundancy problem in database tables using normalization.
CO 4:	Understand the concepts of transactions, their processing so they will know with broad range of database management issues together with data integrity, security and recovery.
CO 5:	Design, develop and implement a small database project using database tools.

Unit	Topic
I	**Introduction:** An overview of database management system, database system vs. file system, Database system concept an architecture, data model schema and instances, data independence and database language and interfaces, data definitions language, DML, Overall Database Structure. **Data Modeling using the Entity Relationship Model:** ER model concepts, notation for ER diagram, mapping constraints, keys, Concepts of Super Key, candidate key, primary key, Generalization, aggregation, reduction of an ER diagrams to tables, extended ER model, relationship of higher degree.
II	**Relational data Model and Language:** Relational data model concepts, integrity constraints, entity integrity, referential integrity, Keys constraints, Domain constraints, relational algebra, relational calculus, and tuple and domain calculus. **Introduction on SQL:** Characteristics of SQL, advantage of SQL. SQL data type and literals, Types of SQL commands, SQL operators and their procedure, Tables, views and indexes, Queries and sub queries, Aggregate functions. Insert, update and delete operations, Joins, Unions, Intersection, Minus, Cursors, Triggers and Procedures in SQL/PLSQL.
III	**Data Base Design and Normalization:** Functional dependencies, normal forms, first, second, third normal forms, BCNF, inclusion dependence, loss less join decompositions, normalization using FD, MVD, and JDs, alternative approaches to database design.
IV	**Transaction Processing Concept:** Transaction system, Testing of serializability, serializability of schedules, conflict and view serializable schedule, recoverability, Recovery from transaction failures, log-based recovery, checkpoints, deadlock handling.
V	**Distributed Database:** distributed data storage, Concurrency Control Techniques: Concurrency control, Locking Techniques for concurrency control, Time stamping protocols for concurrency control, validation-based protocol, multiple granularities, Multi version schemes, Recovery with concurrent transaction, case study of Oracle.

Text Book (s):

1. “Database System Concepts”, Abraham Silberschatz, Henry F. Korth, S. Sudarshan, McGraw-Hill.
2. “Principles of Database and Knowledge–Base Systems”, J. D. Ullman, Computer Science Press.
3. “Fundamentals of Database Systems”, by R. Elmasri and S. Navathe, Pearson Education
4. “Foundations of Databases”, Reprint by Serge Abiteboul, Richard Hull, Victor Vianu, Addison-Wesley

TECHNICAL COMMUNICATION
(THM401)

Course Outcome (CO):	
CO 1:	Students will be enabled to understand the nature and objective of Technical Communication relevant for the work place as Engineers and reinforcement learning, and their applications in solving real-world problems.
CO 2:	Students will utilize the technical writing for the purposes of Technical Communication and its exposure in various dimensions.
CO 3:	Students would imbibe inputs by presentation skills to enhance confidence in face of diverse audience.
CO 4:	Technical communication skills will create a vast know-how of the application of the learning to promote their technical competence.
CO 5:	It would enable them to evaluate their efficacy as fluent and efficient

Unit	Topic
I	**Fundamentals of Technical Communication:** Technical Communication: Features; Distinction between General and Technical Communication; Language as a tool of Communication; Dimensions of Communication: Reading and comprehension; Technical writing: sentences; Paragraph; Technical style: Definition, types and Methods; The flow of Communication: Downward; upward,
II	**Forms of Technical Communication:** Technical Report: Definition and importance; Thesis/Project writing: structure and importance; synopsis writing: Methods; Technical research Paper writing: Methods and style; Seminar and Conference paper writing; Expert Technical Lecture: Theme clarity; Analysis and Findings; 7 Cs of effective business writing: concreteness, completeness, clarity, conciseness, courtesy, correctness, consideration, C.V./Resume writing; Technical Proposal: Types, Structure and Draft.
III	**Technical Presentation: Strategies and Techniques:** Presentation: Forms; interpersonal Communication; Class room presentation; style; method; Individual conferencing: essentials: Public Speaking: method; Techniques: Clarity of substance; emotion; Humour; Modes of Presentation; Overcoming Stage Fear; Audience Analysis and retention of audience interest; Methods of Presentation: Interpersonal; Impersonal; Audience Participation: Quizzes and
IV	**Technical Communication Skills:** Interview skills; Group Discussion: Objective and Method; Seminar/Conferences Presentation skills: Focus; Content; Style; Argumentation skills: Devices: Analysis; Cohesion and Emphasis; Critical thinking; Nuances: Exposition narration and Description; effective business communication competence: Grammatical; Discourse competence: combination of expression and conclusion; Socio-linguistic competence: Strategic competence: Solution of communication problems with
V	**Dimensions of Oral Communication and Voice Dynamics:** Code and Content; Stimulus and Response; Encoding process; Decoding process; Pronunciation Etiquette; Syllables; Vowel sounds; Consonant sounds; Tone: Rising tone; Falling Tone; Flow in Speaking; Speaking with a purpose; Speech and personality; Professional Personality Attributes: Empathy; Considerateness; Leadership;

Text Book (s):

1. Technical Communication – Principles and Practices by Meenakshi Raman and Sangeeta Sharma, Oxford Univ. Press, 2007, New Delhi.
2. Personality Development and Soft Skills by Barun K. Mitra, OUP, 2012, New Delhi.
3. Spoken English- A Manual of Speech and Phonetics by R. K. Bansal and J. B. Harrison, Orient Blackswan, New Delhi.

DESIGN AND ANALYSIS OF ALGORITHMS
(TCS451)

LIST OF THE EXPERIMENTS:

1. Program for Recursive Binary and Linear Search.

2. Program for Heap Sort.

3. Program for Merge Sort.

4. Program for Selection Sort.

5. Program for Insertion Sort.

6. Program for Quick Sort.

7. Knapsack Problem using Greedy Solution

8. Perform Travelling Salesman Problem

9. Find Minimum Spanning Tree using Kruskal's Algorithm

10. Implement N Queen Problem using Backtracking

11. Sort a given set of n integer elements using Quick Sort method and compute its time complexity. Run the program for varied values of n> 5000 and record the time taken to sort. Plot a graph of the time taken versus non graph sheet. The elements can be read from a file or can be generated using the random number generator. Demonstrate using Java how the divide and- conquer method works along with its time complexity analysis: worst case, average case and best case.

12. Sort a given set of n integer elements using Merge Sort method and compute its time complexity. Run the program for varied values of n> 5000, and record the time taken to sort. Plot a graph of the time taken versus non graph sheet. The elements can be read from a file or can be generated using the random number generator. Demonstrate how the divide andconquer method works along with its time complexity analysis: worst case, average case and best case.

13. Implement, the 0/1 Knapsack problem using

(a) Dynamic Programming method

(b) Greedy method.

14. From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm.

15. Find Minimum Cost Spanning Tree of a given connected undirected graph using Kruskal's algorithm. Use Union-Find algorithms in your program.

16. Find Minimum Cost Spanning Tree of a given undirected graph using Prim’s algorithm.

17. Write programs to

(a) Implement All-Pairs Shortest Paths problem using Floyd's algorithm.

(b) Implement Travelling Sales Person problem using Dynamic programming.

18. Design and implement to find a subset of a given set S = {Sl, S2,.....,Sn} of n positive integers whose SUM is equal to a given positive integer d. For example, if S ={1, 2, 5, 6, 8} and d= 9, there are two solutions {1,2,6}and {1,8}. Display a suitable message, if the given problem instance doesn't have a solution.

19. Design and implement to find all Hamiltonian Cycles in a connected undirected Graph G of n vertices using backtracking principle.

Note: The instructor may add/ modify the list as per his/her wisdom for better hands on the students.

OBJECT ORIENTED PROGRAMMIING WITH JAVA LAB
(TCS452)

LIST OF THE EXPERIMENTS:

1. Use Java compiler and eclipse platform to write and execute java program.
2. Creating simple java programs using command line arguments
3. Understand OOP concepts and basics of Java programming.
4. Create Java programs using inheritance and polymorphism.
5. Implement error-handling techniques using exception handling and multithreading.
6. Create java program with the use of java packages.
7. Construct java program using Java I/O package.
8. Create industry-oriented application using Spring Framework.
9. Test RESTful web services using Spring Boot.
10. Test Frontend web application with Spring Boot

Note: The instructor may add/ modify the list as per his/her wisdom for better hands on the students.

COMPUTER ORGANISATION and ARCHITECTURE LAB
(TCS453)

LIST OF THE EXPERIMENTS:

- Implementing Half-Adder, Full-Adder using basic logic gates
- Implementing Binary -to -Gray, Gray -to -Binary code conversions.
- Implementing 3–8-line Decoder.
- Implementing 4x1 and 8x1 Multiplexers.
- Verify the excitation tables of various Flip-Flops.
- Design of an 8-bit Input/ Output system with four 8-bit Internal Registers.
- Design of an 8-bit Arithmetic Logic Unit.
- Design the data path of a computer from its register transfer language description.
- Implement a simple instruction set computer with a control unit and a data path.

Note: The instructor may add/ modify the list as per his/her wisdom for better hands on the students.

DATABASE MANAGEMENT SYSTEM LAB
(TCS554)

LIST OF THE EXPERIMENTS:

1. Write the queries for Data Definition and Data Manipulation Language.
2. Write SQL queries using logical operators.
3. Write SQL queries using SQL operators
4. Write SQL query using character, number, date and group functions
5. Write SQL queries for relational algebra
6. Write SQL queries for extracting data from more than one table
7. Write SQL queries for sub queries, nested queries
8. Write program of PL/SQL
9. Concepts for ROLL BACK, COMMIT and CHECK POINTS
10. Create VIEWS, CURSORS and TRGGERS

Note: The instructor may add/ modify the list as per his/her wisdom for better hands on the students.