SCHEME OF INSTRUCTION AND DETAILED SYLLABI FOR

MAJOR DEGREE

B. TECH

IN

INFORMATION TECHNOLOGY

Effective from 2023 Batch onwards

Dr. B R AMBEDKAR NATIONAL INSTITUTE OF TECHNOLOGY

JALANDHAR – 144008

Year Wise Credit Distribution

Year	No of Credits
I	45
II	44
III	46
IV	29
Total	**164**

Semester – I

S. No.	Course Code	Subject	L	T	P	Credits	Contact Hours
1.	MAFC0101	Mathematics-I	3	1	0	4	4
2.	EEFC0101	Basics of Electrical Engineering	3	0	0	3	3
3.	ITDC0101	Probability and Computing	3	1	0	4	4
4.	CSFC0101	Computer Programming	3	0	0	3	3
5.	HMFC0101	English Communication and Report Writing	2	0	0	2	2
6.	IDFC0101	Environmental Sciences	3	0	0	3	3
7.	EEFC0131	Basics of Electrical Engineering Lab	0	0	2	1	2
8.	CSFC0131	Computer Programming Lab	0	0	2	1	2
9.	HMFC0131	English Communication Lab	0	0	2	1	2
10.	IDFC0101	Environmental Sciences Lab	0	0	2	1	2
11.	PDFE01XX	Foundation Elective-1	-			1	3
		Total				**24**	**30**

Semester – II

S. No.	Course Code	Subject	L	T	P	Credits	Contact Hours
1.	MAFC0102	Mathematics-II	3	1	0	4	4
2.	IPFC0101	Engineering Graphics and CAD	2	0	4	4	6
3.	ITFC0101	Problem-Solving using Python	2	0	0	2	2
4.	ITDC0102	Introduction to Digital Circuits and Communications	3	1	0	4	4
5.	HMFC0102	Management Principles and Indian Constitutional Values	3	0	0	3	3
6.	ITFC0131	Problem-Solving using Python Lab	0	0	2	1	2
7.	WPFC0131	Product Realization through Manufacturing Processes (Workshop Practice)	0	0	4	2	4
8.	PDFE01XX	Foundation Elective-2	-			1	3
		Total				**21**	**28**

Semester – III

S. No.	Course Code	Subject	L	T	P	Credits	Contact Hours
1.	ITDC0201	Object Oriented Programming	3	0	0	3	3
2.	ITDC0203	Data Structures	3	0	0	3	3
3.	ITDC0205	Web Technologies	3	0	0	3	3
4.	ITDC0207	Database Management Systems	3	0	0	3	3
5.	ITDC0209	Computer System Architecture	3	1	0	4	4
6.	ITDC0211	Discrete Structures	3	0	0	3	3
7.	ITDC0231	Object Oriented Programming Lab	0	0	2	1	2
8.	ITDC0233	Data Structures Lab	0	0	2	1	2
9.	ITDC0235	Web Technologies Lab	0	0	2	1	2
10.	ITDC0237	Database Management Systems Lab	0	0	2	1	2
		Total				**23**	**27**

Semester – IV

S. No.	Course Code	Subject	L	T	P	Credits	Contact Hours
1.	ITDC0202	Design and Analysis of Algorithms	3	0	0	3	3
2.	ITDC0204	Operating System	3	0	0	3	3
3.	ITDC0206	Software Engineering	3	0	0	3	3
4.	ITDC0208	Computer Networks	3	0	0	3	4
5.	ITDC0210	Formal Language and Automata Theory	3	1	0	4	4
6.	ITFE0202	Competitive Programming (FE-III)	1	0	0	1	1
7.	ITDC0232	Design and Analysis of Algorithms Lab	0	0	2	1	2
8.	ITDC0234	Operating System Lab	0	0	2	1	2
9.	ITDC0236	Software Engineering Lab	0	0	2	1	2
10.	ITDC0238	Computer Networks Lab	0	0	2	1	2
		Total				**21**	**26**

Semester – V

S. No.	Course Code	Subject	L	T	P	Credits	Contact Hours
1.	ITDC0301	Object Oriented Modeling and Design using UML	3	0	0	3	3
2.	ITDC0303	Java Programming	3	0	0	3	3
3.	ITDC0305	Artificial Intelligence	3	0	0	3	3
4.	ITDC0307	Cryptography and Network Security	3	0	0	3	3
5.	ITDE03xx	Department Elective-I	3	0	0	3	3
7.	ITDC0333	Java Programming Lab	0	0	2	1	2
8.	ITDC0335	Artificial Intelligence Lab	0	0	2	1	2
9.	ITDC0337	Cryptography and Network Security Lab	0	0	2	1	2
11.	ITTR0300	Industrial Training (After 4th Sem.)	-`	-	-	2	-
12.	ITPR0301	Project-1 (Phase-I)	0	0	4	2	2
		Total				**22**	**23**

Semester – VI

S. No.	Course Code	Subject	L	T	P	Credits	Contact Hours
1.	ITDC0302	Distributed System and Cloud Computing	3	0	0	3	3
2.	ITDC0304	Full Stack Web Development	3	0	0	3	3
3.	ITDC0306	Big Data Analytics	3	0	0	3	3
4.	ITDC0308	Machine Learning	3	0	0	3	3
5.	ITDE03xx	Department Elective-II	3	0	0	3	3
6.	ITGE03xx	Generic Elective -I	3	0	0	3	3
7.	ITDC0332	Distributed System and Cloud Computing Lab	0	0	2	1	2
8.	ITDC0334	Full Stack Web Development Lab	0	0	2	1	2
9.	ITDC0336	Big Data Analytics Lab	0	0	2	1	2
10.	ITDC0338	Machine Learning Lab	0	0	2	1	2
11.	ITPR0302	Project-1 (Phase-II)	0	0	4	2	2
		Total				**24**	**28**

Semester – VII							
S. No.	**Course Code**	**Subject**	**L**	**T**	**P**	**Credits**	**Contact Hours**
1.	ITDE04xx	Department Elective-III	3	0	0	3	3
2.	ITDE04xx	Department Elective-IV	3	0	0	3	3
3.	ITGE0451	Generic Elective-II	3	0	0	3	3
4.	ITTR0400	Industrial Training (After 6th Sem.)	-	-	-	3	1
5.	ITPR0401	Project-2[1]	0	0	8	4	8
		Total				**16**	**17**
Semester – VIII							
S. No.	**Course Code**	**Subject**	**L**	**T**	**P**	**Credits**	**Contact Hours**
1.	ITDE04xx	Department Elective-V	3	0	0	3	3
2.	ITDE04xx	Department Elective-VI	3	0	0	3	3
3.	ITGE04xx	Generic Elective-III	3	0	0	3	3
5.	ITPR0402	Project-2[2]	0	0	8	4	8
		Total				**13**	**27**

LIST OF DEPARTMENTAL ELECTIVES (taught to department students)

DEPARTMENTAL ELECTIVES I

S.No.	Subject Code	Subject Name	L	T	P	Credits
1	ITDE0351	Data Mining	3	0	0	3
2	ITDE0353	Computer Graphics and Applications	3	0	0	3

DEPARTMENTAL ELECTIVES II

S.No.	Subject Code	Subject Name	L	T	P	Credits
1	ITDE0352	Mobile Communication	3	0	0	3
2	ITDE0354	Principles of Programming Languages	3	0	0	3
3	ITDE0356	Business Intelligence	3	0	0	3

DEPARTMENTAL ELECTIVES III

S.No.	Subject Code	Subject Name	L	T	P	Credits
1	ITDE0451	Cyber Security	3	0	0	3
2	ITDE0453	Soft Computing	3	0	0	3
3	ITDE0455	Internet of Things (IoT)	3	0	0	3
4	ITDE0457	Game Design *(MOOC)*	3	0	0	3

DEPARTMENTAL ELECTIVES IV

S.No.	Subject Code	Subject Name	L	T	P	Credits
1	ITDE0461	Wireless Communication for Beyond 5G Networks and IoT	3	0	0	3
2	ITDE0463	Image Processing	3	0	0	3
3	ITDE0465	Prompt Engineering	3	0	0	3
4	ITDE0467	Ethical Hacking *(MOOC)*	3	0	0	3

DEPARTMENTAL ELECTIVES V

S.No.	Subject Code	Subject Name	L	T	P	Credits
1	ITDE0452	DevOps/MLOps	3	0	0	3
2	ITDE0454	Mobile Application Development	3	0	0	3
3	ITDE0456	High Performance Computing	3	0	0	3

DEPARTMENTAL ELECTIVES VI

S.No.	Subject Code	Subject Name	L	T	P	Credits
1	ITDE0462	Natural Language Processing	3	0	0	3
2	ITDE0464	Computer Vision	3	0	0	3
3	ITDE0466	Federated Learning	3	0	0	3

LIST OF GENERIC ELECTIVES (taught to other departments)

GENERIC ELECTIVES I

S.No.	Subject Code	Subject Name	L	T	P	Credits
1	ITGE0351	Web Design Concepts	3	0	0	3
2	ITGE0353	Agile Software Engineering	3	0	0	3

GENERIC ELECTIVES II

S.No.	Subject Code	Subject Name	L	T	P	Credits
1	ITGE0352	Fundamentals of Data Analytics	3	0	0	3
2	ITGE0354	Fundamentals of Cloud Computing	3	0	0	3

GENERIC ELECTIVES III

S.No.	Subject Code	Subject Name	L	T	P	Credits
1	ITGE0451	Machine Learning	3	0	0	3
2	ITGE0453	Fundamental of Artificial Intelligence	3	0	0	3

GENERIC ELECTIVES IV

S.No.	Subject Code	Subject Name	L	T	P	Credits
1	ITGE0452	Mobile Application Development Concepts	3	0	0	3
2	ITGE0454	Low Code/No Code Technology	3	0	0	3

Semester-I

Course Code	**ITDC0101**
Subject Name	**PROBABILITY AND COMPUTING**
Contact hours/Credit Scheme (L-T-P-C)	**3-1-0-4**
Pre-requisites	**None**

Course Objective:

To course aims to develop the understanding of basic concepts in probability theory that are relevant to computing. The course accompanies the insights mapping of computer science and applied mathematics enabling students to have foundations for mathematical computations.

Course Outcomes:

CO 1. Understand the fundamental concepts of basic probability, conditional probability, events, and Bayes' theorem to solve the mathematical problems.
CO 2. Solve and analyze the real problems using properties of random processes, marginal and conditional probability distributions, Markov chain, and correlation of random variables.
CO 3. Compare the performance of multiple methods and models to recognize the connections between basic inference and parametric inference with the help of confidence interval and moment estimator.
CO 4. Illustrate the utility of various methods of hypothesis testing in multiple regression models to measure the performance and fitness in real-world problems.

Course Outcomes	**Program Outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDC0101														
CO 1	H			L					M			M		H
CO 2	L	L		M			M		M		M	M		M
CO 3		H	M	H	M				M					L
CO 4	M	M	M	H	M	L	M		M		M	M	L	L

Course Content:

Unit 1: Basics: sample space, outcomes, probability, Conditional Probability, Events: mutually, exclusive, independent, calculating probability: sets, counting, tree diagram, Conditional probability, Law of total probability, Bayes' theorem, Random Process, Finite Markov's chain, Markov's process, Poisson Processes, M/M/1- finite and infinite, birth-death processes

Unit 2: Mean, Moments, Variance, PMF, PDF, CDF, Bernoulli(p), Indicator RV, Binomial (n, p), Geometric(p), Uniform (a, b), Exponential(λ), Normal (μ, $\sigma 2$), and its several properties, Joint probability distribution, Linearity and product of expectation

Unit 3: Central Limit Theorem, Basics of inference, Empirical PMF, Sample mean, bias, se, MSE, Empirical Distribution Function (or eCDF), Statistical Functionals, Plug-in estimator, Confidence intervals: Percentiles, quantiles, Normal-based confidence intervals, DKW inequality, Basics of parametric inference, Method of Moments Estimator (MME), Method of Moments Estimator (MME), Properties of MME, Likelihood, Maximum, Likelihood Estimator (MLE), Properties of MLE

Unit 4: Hypothesis testing, Basics of hypothesis testing, The Wald test, Type I, and Type II errors, t-test, Kolmogorov-Smirnov test (KS test), p-values, Permutation test, Pearson correlation coefficient, Chi-square test for independence, Bayesian reasoning, Bayesian inference, Bayesian inference, Conjugate priors, Basics of Regression, Simple Linear Regression, Multiple Linear Regression.

Text/Reference Books:

1. Kris Harmans "Mastering Probability and Statistics: A Comprehensive Guide to Learn Probability and Statistics" First Edition, Cybellium Ltd, 2023.
2. Krishna Kumar Chaudhary, Anjay Kumar Mishra, Raju Ram Thapa, "A Reference Book on Integrated probability and statistics", First Edition, Intellectuals' book palace , 2022
3. Mastering Probability and Statistics: A Comprehensive Guide to Learn Probability and Statistics, 2023.
4. Durrett, Rick. Probability: theory and examples. Vol. 49. Cambridge university press, 2019.
5. Kishor S. Trivedi, "Probability and Statistics with Reliability, Queuing and Computer Science Applications", John Wiley & Sons, 2016
6. Durrett, Rick. Probability: theory and examples. Vol. 49. Cambridge university press, 2019.
7. Gómez-Rubio, Virgilio. Bayesian inference with INLA. CRC Press, 2020.
8. Hartshorn, Scott. Hypothesis testing: a visual introduction to statistical significance. 2017.
9. Verma, J. P., and Abdel-Salam G. Abdel-Salam. Testing statistical assumptions in research. John Wiley & Sons, 2019.

Course Code	**ITDC0102**
Subject Name	**DIGITAL CIRCUITS AND DIGITAL COMMUNICATIONS**
Contact hours/Credit Scheme (L-T-P-C)	**3-1-0-4**
Pre-requisites	**None**

Course Objective:

To familiarize understanding of computer network basic, different models used for study of computer networks, ability to identify different designs, understanding of the issues surrounding wired and wireless Networks.

Course Outcomes:

CO 1. Familiarize with the fundamentals of the number systems, binary operations and floating point representation.
CO 2. Apply logic gates and Karnaugh maps to minimize Boolean identities.
CO 3. Analyze and design combinational/sequential logic circuits using logic gates and flip flops.
CO 4. Analyze the scope of digitization, encoding, conversion and transmission of a signal in digital communication.

Course Outcomes	**Program Outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDC0102														
CO 1	H										L	L		M
CO 2		H			H									
CO 3			H	L	H		L				L		M	
CO 4			M	M		M	H		L		L	M		M

Course Content:

Unit 1: Number Systems- Decimal, Binary, Octal, Hexadecimal and their inter-conversions, Representation of Data: Signed Magnitude, one's complement and two's complement, Binary Arithmetic, Fixed point representation and Floating point representation of numbers.

Unit 2: Boolean Algebra- Basic gates (AND, OR, NOT gates), Universal gates (NAND and NOR gates), other gates (XOR, XNOR gates). Boolean identities, De Morgan Laws, Karnaugh Map (K-map).

Unit 3: Combinational and sequential circuits- Half adder, full adder, code converters, combinational circuit design, Multiplexers and demultiplexers, encoders, decoders, Combinational design using mux and demux. Sequential circuits- Flip-flops, counters, and their types, shift registers.

Unit 4: Data Communication- Data Transmission/The Physical Layer: Concepts and Terminology, Analog and Digital Data Transmission, Transmission Impairments, Guided Transmission Media, Wireless Transmission, Communication Satellites.
Data Encoding: Digital Data: Digital and Analog Signals, Analog Data: Digital and Analog Signals
Data Communication Interface: Asynchronous and Synchronous Transmission, Line Configurations, Interfacing
Multiplexing: Frequency-Division Multiplexing, Synchronous Time-Division Multiplexing, Statistical Time- Division Multiplexing
Circuit Packet and Switching: Switched Networks, Circuit-Switching Networks, Switching Concepts, Routing in Circuit-Switched Networks, Control Signaling, Packet-Switching Principles, Routing, Congestion Control.

Text/Reference Books:

1. Salivahanan, S., and S. Arivazhagan. Digital circuits and design. Vikas publishing house PVT Limited. Fifth edition (March 2018)
2. Analog and Digital Communication by S. Rameshbabu, Dr. S. Prabhakar rao, J. Sunilkumar, Sahebgoud karaddi. 2022
3. Digital Communication Techniques by Christian Gontrand. 2020
4. Oppenheim, Alan V., Alan S. Willsky, and Syed Hamid Nawab. "Signals and Systems," Prentice-Hall Englewood Cliffs 2 edition (2015)
5. Andrew S. Tanenbaum, "Computer Networks" Ed Pearson Education 4th Edition, 2010.
6. Behrouz A. Fourouzan, Data Communications, and Networking, 2/e Tata McGraw-Hill, 2006.
7. S. Haykin, Digital Communications Systems, John Wiley & Sons, 1st edition, 2013.

Course Code	**ITFC0101**
Subject Name	**PROBLEM SOLVING USING PYTHON**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**None**

Course Objective:

To develop overview and understanding of the key concepts of Python programming language, principles of computers, binary computation, various data types and control structure, usage of the object-oriented program design and development and inculcate the practical implementation.

Course Outcomes:

CO 1. Understand the fundamental concepts of Python programming and its libraries to develop basic programs.
CO 2. Develop skills to design, implement and debug programs using modular programming.
CO 3. Devise a real time system by applying various concepts of python including basic data structures and file systems.
CO 4. Apply the learned concepts and design GUI based applications in Python.

Course Outcomes	**Program Outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITFC0101														
CO 1	H	M	L									M		L
CO 2		H		M	L	M							M	

CO 3			M		H	M			L				L	M
CO 4			M	H			M		H			H		

Course Content:

Unit 1: Computational Thinking and Problem-Solving: Identification of Computational Problems -Algorithms, building blocks of algorithms (statements, state, control flow, functions), notation (pseudo code, flow chart, programming language), algorithmic problem solving, simple strategies for developing algorithms (iteration, recursion).

Unit 2: Data Types, Expressions, and Statements: Python interpreter and interactive mode, debugging; values and types: int, float, Boolean, string, and list; variables, expressions, statements, tuple assignment, precedence of operators, comments.

Unit 3: Control Flow Functions, and Strings: Conditions - Boolean values and operators, conditional (if), alternative (if-else), chained conditional (if-elif-else); Iteration - state, while, for, break, continue, pass; Fruitful functions - return values, parameters, local and global scope, function composition, recursion; Strings - string slices, immutability, string functions and methods, string module; Lists as arrays.

Unit 4: Lists, Tuples, Dictionaries: Lists - list operations, list slices, list methods, list loop, mutability, aliasing, cloning lists, list parameters; Tuples - tuple assignment, tuple as a return value; Dictionaries - operations and methods; advanced list processing, list comprehension.

Unit 5: Files, Modules, Packages: Files and exception: text files, reading and writing files, format operator; command line arguments, errors and exceptions, handling exceptions, modules, packages

Text/Reference Books:

1. Paul Deitel and Harvey Deitel, “Python for Programmers”, Pearson Education, 1st Edition, 2021.
2. Puntambekar, Anuradha A., “Programming and Problem Solving using Python”. Second Edition, Repro Books Limited, 2020.
3. Wesley J. Chun, “Core Python Applications Programming”, 3rd Edition , Pearson Education, 2016.
4. Charles Dierbach, “Introduction to Computer Science using Python”, Wiley, 2015.
5. Jeeva Jose &P.SojanLal, “Introduction to Computing and Problem Solving with PYTHON”, KhannaPublishers, New Delhi, 2016.
6. Downey, A. et al., “How to think like a Computer Scientist: Learning with Python”, John Wiley, 2015.
7. David Beazley, Brian Jones., “Python Cookbook”, Third Edition, Orelly Publication, 2013, ISBN 978-1449340377
8. Michel Dawson, “Python Programming for Absolute Beginers” , Third Edition, Course TechnologyCengage Learning Publications, 2013, ISBN 978-1435455009

Course Code:	**ITFC0131**
Subject Name	**PROBLEM SOLVING USING PYTHON LAB**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**None**

Course Objective:

To develop the implementation skills related to the key concepts of Python programming language, principles of computers, binary computation, various data types and control structure, usage of the object-oriented program design and development and apply these for the real-world situations.

Course Outcomes:

CO 1. Understand the fundamental concepts of Python programming and its libraries to develop basic programs.
CO 2. Develop skills to design, implement and debug programs using modular programming.
CO 3. Devise a real time system by applying various concepts of python including basic data structures and file systems.
CO 4. Apply the learned concepts and design GUI based applications in Python.

Course Outcomes	**Program Outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2

ITFC0131														
CO 1	H	M	L									M		L
CO 2		H		M	L	M							M	
CO 3			M		H	M			L				L	M
CO 4			M	H			M		H			H		

List of Practicals:

1. Exploring Python basics and setting up the development environment
2. Implementing basic Python programs for demonstrating the concepts of python data types, operators and different data structures e.g. find minimum in a list, insert a card in a list of sorted cards, guess an integer number in a range, Towers of Hanoi.
3. Implement programs using expressions and statements e.g. exchange the values of two variables, circulate the values of n variables, distance between two points.
4. Implementing functions in Python and using inbuilt modules
5. Creating and importing user-defined modules in Python
6. Writing python programs for showing the use of anonymous and inner functions.
7. Develop programs using modular programming e.g. square root, gcd, exponentiation, sum an array of numbers, linear search, binary search.
8. Design and implement programs using basic data structures e.g. simple sorting, histogram, Students marks statement, Retail bill preparation.
9. Data manipulation using Pandas library: Reading and writing data, manipulating data using various operations
10. Data handling using files: word count, copy file, Voter's age validation, Marks range validation (0-100).

**This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents*

Semester-III

Course Code	**ITDC0201**
Subject Name	**OBJECT ORIENTED PROGRAMMING**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**None**

Course Objectives:

1. Able to differentiate between structure oriented programming and object oriented programming and to specify simple abstract data types and design implementations
2. Recognize features of object oriented design such as encapsulation, polymorphism, inheritance and composition of systems based on object identity.
3. Able to use object oriented programming language like C++ and associated library to develop object oriented programs.

Course Outcomes:

CO 1. Describe the procedural and object oriented paradigm with concepts of streams, classes,

functions, data and objects.

CO 2. Understand dynamic memory management techniques using pointers, constructors, destructors, etc.

CO 3. Describe the concept of function overloading, operator overloading, virtual functions and polymorphism.

CO 4. Demonstrate the use of various object oriented Programming concepts with the help of programs.

Course Outcomes	Program Outcomes													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDC0201														
CO 1	M	M	H	L	L	L				M			M	M
CO 2	M	M	H	L	L					M			M	M
CO 3	M	L	H	M	L		M			M			M	M
CO 4	L	M	H	M					M	L			M	M

Course Content:

Unit 1: Object Oriented Programming: Need for object-oriented programming paradigm, Procedural programming Vs object-oriented programming, object-oriented concepts. Functions: Main function, function prototyping, inline functions, reference variables, call by reference, Defaults arguments, function overloading, Math library functions.

Unit 2: Class: Difference between C structure and class, specifying a class, member functions, scope resolution operator, Array within a class, array of objects, Static data members and member functions, Friend function.

Unit 3: Constructor and destructor: Constructor, Types of constructors, dynamic initialization of objects, Destructor, Operator overloading and Type Conversion: Defining operator overloading, overloading unary and binary operator. Introduction to Pointer, Pointer and Address, The Address of Operator &, Pointer Variable, Void Pointer, Dangling Pointer, Null Pointer

Unit 4: Inheritance, Polymorphism and Pointer: Base class, derived class, visibility modes, derivation and friendship, Types of inheritance, Containership, virtual function binding, pure virtual functions, Abstract class, pointer to derived class.

Unit 5: Console IO operations: Unformatted IO operations, formatted IO operations, managing output with manipulators. Working with files, Error handling during file operations, Templates.

Chatgpt and youtube

Text/Reference Books:

1. Sunil Kumar Saini, Object-Oriented Programming Understanding Classes and Objects : Practical Tips and Techniques for Object-Oriented Programming, Drinkin Go, 2023.
2. Yashwant Kenetkar, “Let us C++”,19th Ed., Oxford University Press, 2022.
3. Richard Baker, Object Oriented Programming in C++: Object Oriented Programming & Features of OOP's, Independently Published, 2020.
4. Bjarne Stroustrup: The C++ Programming Language (4th Edition). Addison-Wesley. ISBN 978-

0321563842. May 2013.
5. B.A. Forouzan and R.F. Gilberg, “Computer Science: A structured approach using C++” 3rd Ed. Cengage Learning, New Delhi, 2006.
6. E. Balagurusamy, “Object Oriented Programming with C++” 8th Ed., Tata McGraw Hill, 2020.
7. C++: THE COMPLETE REFERENCE Paperback – 1 July 2017 by Herbert Schildt (Author)

Course Code	**ITDC0203**
Subject Name	**DATA STRUCTURES**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**None**

Course Objectives:

1. This course introduces the concept of Data Structures used in various computer science applications.
2. The students are introduced to understand and efficiently apply various data structures such as stacks, queues, linked lists, trees and graphs for solving various computing problems using C programming language.

Course Outcomes:

CO1. Develop skill to identify and determine the usage of various data structures, operations, associated algorithm and implement their applications.
CO2. Apply knowledge of pointers, memory allocation and string handling for solving programming problems.
CO3. Understand the concept of trees and graphs, their implementation and applications.
CO4. Able to implement standard algorithms for searching and sorting.
CO5. Analyze efficiency of different algorithms using time and space complexity.

Course Outcomes	**Program Outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDC0203														
CO 1	H	M	M	M	M								H	
CO 2			M		M	M	M		L					H
CO 3	L	M	M	H	M								M	
CO 4		H	H	L									M	
CO 5	M	M	H		M	H	M		H		H			M

Course Content:

Unit 1: Introduction: Basic Terminology, Elementary Data Organization, Algorithm, Efficiency of an Algorithm, Time and Space Complexity, Asymptotic notations: Big-Oh, Time-Space trade-off. Abstract Data Types (ADT),

Unit 2: Array and Linked lists: Arrays: Definition, Single and Multidimensional Arrays, Representation of Arrays: Row Major Order, and Column Major Order, Application of arrays, Sparse Matrices and their representations Singly Linked Lists, Doubly Linked List, Circularly Linked List, Operations on a Linked List. Insertion, Deletion, Traversal, Applications.

Unit 3: Stack and Queues: Stacks: Introduction, Stack operations: Push & Pop, Prefix and Postfix Expressions, Applications, Operations on Queue: Create, Add, Delete, Full and Empty, Circular queues, Dequeue, Priority Queue, Applications.

Unit 4: Trees: Basic Terminology, Binary Trees, Binary Tree Representation, Complete Binary Tree, Tree Traversal algorithms, Huffman algorithm. Search Trees: Binary Search Trees (BST), Insertion and Deletion in BST, Complexity of Search Algorithm, AVL trees, Introduction to m-way Search Trees, B

Trees & B+ Trees, Graphs: Terminology, Representations of Graphs, Graph Traversal: Depth First Search and Breadth First Search, Connected Component, Spanning Trees, Minimum Cost Spanning Trees: Prims and Kruskals algorithm.

Unit 5: Searching and Sorting: Sequential Search, Binary Search, Comparison and Analysis, Insertion Sort, Selection Sort, Bubble Sort, Quick Sort, Heap Sort, Radix Sort. Hash Tables: Basic Hash Tables and Collision Resolution Techniques.

Text/Reference Books:

1. Schaum Outline series by: Seymour Lipschutz, "Data Structures", Publishers: Tata McGraw Hill, New Delhi, Year of Publication: 2017.
2. M. Tenenbaum, "Data Structures using C & C++", Prentice-Hall of India Pvt. Ltd., New Delhi, 2nd Ed., Year of Publication:2015.
3. E. Horowitz, S. Sahni, S.Anderson-freed, "Fundamentals of Data Structures in C", Second Edition, University Press, 2017.
4. R. Kruse and C.L. Tondo, "Data Structures and Program Design in C", 2nd Edition, Pearson, Year of Publication: 2006
5. Gilberg Forozan, "Data Structure – A pseudo code approach with C++", Cengage Learning, New Delhi. 2nd Edition, Year of Publication: 2015
6. Peter Brass, "Advance Data Structure", Cambridge University Press, First Edition, Year of Publication: 2016
7. James A. Storer, "An Introduction to Data Structure and Algorithms", Springer Science & Business Media, Year of Publication: 2012

Course Code	**ITDC0205**
Subject Name	**WEB TECHNOLOGIES**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives:

1. Introduction and brief history of world wide web (WWW).
2. Web essentials: HTML, XHTML, CSS.
3. Addressing web standards, audience requirements and principles of web page design.
4. Introduction of Web architecture, databases, jdbc.

Course Outcomes:

CO1. Understand basic principles of web site design, considering the information architecture.
CO2. Incorporate best practices in navigation, usability in website design
CO3. Design of website adhering to current web standards (HTML, XML, CSS)
CO4. Learning various scripting languages to create interactive components in web pages.
CO5. Analyze different web standards and design websites as per compliance of different standards.

Course Outcomes	Program Outcomes													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDC0205														
CO 1	H		M		M									
CO 2	M	M	H	M	M				H				H	
CO 3					H	M								
CO 4		H		H		H		H		M		M	H	

Course Content:

Cheat sheet and one shot

Unit 1: Introduction to HTML: HTML Common tags- List, Tables, images, forms, Frames; Cascading Style sheets;

Unit 2: Introduction to JavaScript: Scripts, Objects in Java Script, Dynamic HTML with Java Script XML: Document type definition, XML Schemas, Document Object model, Presenting XML, Using XML Processors: DOM and SAX

Chatgpt and pdf

Unit 3: Java Beans: Introduction to Java Beans, Advantages of Java Beans, BDK Introspection, Using Bound properties, Bean Info Interface, Constrained properties Persistence, Customizes, Java Beans API, Introduction to EJB's

Playlist and Chatgpt theory

Unit 4: Web Servers and Servlets: Tomcat web server, Introduction to Servelets: Lifecycle of a Serverlet, JSDK, The Servelet API, The javax.servelet HTTP package, Handling Http Request & Responses, Using Cookies-Session Tracking, Security Issues,

Unit 5: Introduction to JSP: The Problem with Servelet. The Anatomy of a JSP Page, JSP Processing. JSP Application Design with MVC Setting Up and JSP Environment: Installing the Java Software Development Kit, Tomcat Server & Testing Tomcat. JSP Application Development, Conditional Processing, Error Handling and Debugging Sharing Data Between JSP pages, Requests, and Users Passing Control and Date between Pages – Sharing Session and Application Data – Memory Usage Considerations. Database Programming using JDBC, Studying Javax.sql.* package, Accessing a Database from a JSP Page,

Application – Specific Database Actions, Deploying JAVA Beans in a JSP Page, Introduction to struts framework.

Text/Reference Books:

1. Chris Bates,"Web Programming, building internet applications", 3rd Ed., WILEY Dreamtech, 2007.
2. Patrick Naughton and Herbert Schildt, "The complete Reference Java 2" 5th Ed., TMH, 2002.
3. Hans Bergsten , "java Server Pages", 2nd Ed., O'Reilly, 2002.
4. Dietel, Nieto, "Internet and World Wide Web – How to program", 2nd Ed., PHI, 2001.
5. Joel Sklar, "Web Warrier guide to web design technologies", 3rd Ed. Cengage Learning, 2005.

Course Code	**ITDC0207**
Subject Name	**DATABASE MANAGEMENT SYSTEMS**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives:

1. To understand fundamentals of data models and to conceptualize a database system for user requirement.
2. To study the fundamental of Database query language, like SQL and relational algebra.
3. To learn the concept of normalization in database design.
4. To learn fundamental concepts of transaction processing, concurrency control techniques and database recovery procedure.
5. Understand the professional, ethical and security issues and responsibilities in database design.

Course Outcomes:

CO1. To understand fundamental concepts, principles and applications of database system.
CO2. To demonstrate database related programming languages and perform the basics of commercial relational systems.
CO3. To apply the concepts of normalization on database design.
CO4. To Design and Implement a small database project, considering the issues like concurrency control, recovery and security.
CO5. Comprehend contemporary issues related to industry and government related to database domain.

Course Outcomes	**Program Outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDC0207														
CO 1	M		H		M	L							M	
CO 2			H	L	M								M	
CO 3			M	M	M	H	M	M	M					M
CO 4		M	H	L	H	L		H	M			M		M
CO 5						M								M

Course Content:

Unit 1: Introduction: An overview of database management system, database system Vs file system, Database system concepts and architecture, data models schema and instances, data independence and data base 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, relationships of higher degree

Shanu kuttan

Unit 2: Relational data Model and Language: Relational data model concepts, integrity constraints, Keys constraints, Domain constraints, relational algebra. Introduction to SQL: Characteristics of SQL. Advantage of SQL. SQL data types and literals. Types of SQL commands. SQL operators and their procedure. Queries and sub queries. Aggregate functions. Insert, update and delete operations. Joins, Unions, Intersection, Minus, Cursors in SQL.

Shanu kuttan

Gate smashers

Unit 3: Data Base Design & Normalization: Functional dependencies, normal forms, first, second, third normal forms, BCNF, inclusion dependences, loss less join decompositions, normalization using FD, MVD, and JDs, alternative approaches to database design.

Gate smashers

Unit 4: Transaction Processing Concepts: Transaction system, Testing of serializability, Recovery from transaction failures, Recovery concepts based on intermediate update, shadow paging, check points, on-line backup during database updates. Concurrency Control Techniques: Concurrency control, locking Techniques for concurrency control

Chat gpt and gate smashers

Unit 5: Client/Server Databases: Client/Server concepts, approach, architecture, Client/Server communication, APIs in Client/Server computing, Security and Repositories: Needs for database integrity, integrity constraints, non-procedural integrity constraints, integrity constraints specifications in SQL, introduction to database security mechanism, security specification in SQL.

Text/Reference Books:

1. Jagdish Chandra Patni, Hitesh Kumar Sharma, Ravi Tomar, Avita Katal,"Database Management System: An Evolutionary Approach" Third Edition, CRC Press · 2022
2. S. Sureshkumar, Dr. S. Suresh, Mr. S. Joseph James, Mrs. Priya R ,"Database Management System" First Edition, AG Publishing House (2022)
3. Date C J, "An Introduction To Database System", 8th Ed., Addision Wesley, 2013.
4. Korth, Silbertz, Sudarshan, "Database Concepts", 7th Ed., McGraw Hill, 2021.
5. Elmasri, Navathe, "Fundamentals Of Database Systems", 7th Ed., Addision Wesley, 2017.
6. Bipin C. Desai, "An introduction to Database Systems", 8th Ed. (Revised), Galgotia Publication, 2010.
7. Rob, Morris and Coronel, "Database Systems", 10th Ed.,Cengage Learning, 2015.

Course Code	**ITDC0209**
Subject Name	**COMPUTER SYSTEM ARCHITECTURE**
Contact hours/Credit Scheme (L-T-P-C)	**3-1-0-4**
Pre-requisites	**NONE**

Course Objectives:

1. Working of Computer Systems & its basic principles
2. Concepts of I/O devices, processor, control design, and pipelining techniques.
3. Memory hierarchy and its organization.

Course Outcomes:

CO1. Understand the Computer System concepts.
CO2. Understand the organization of a computer system in terms of its main components
CO3. Understand the processor and control design of a computer system.
CO4. Understand the various types of memory.
CO5. Understand the architecture and organization of Computer.

Course Outcomes	**Program Outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDC0209														
CO 1	M	L	M	H			M						M	
CO 2	M	L	L	M	M	M	H						M	
CO 3	H			M		L					H		M	
CO 4	L	M				H					M		M	
CO 5	H		H	M										

Course Content:

Unit 1: Introduction: Number representation; fixed and floating point number representation, IEEE standard for floating point representation. Digital computer generation, computer types and classifications, functional units and their interconnections, buses, bus architecture, types of buses and bus arbitration. Register, bus and memory transfer.

Random Available

Unit 2: Central Processing Unit: Addition and subtraction of signed numbers, look ahead carry adders. Multiplication: Signed operand multiplication, Booths algorithm and array multiplier. Division and logic operations. Floating point arithmetic operation, Processor organization, general register organization, stack organization and addressing modes.

Notes and shanu kuttan

Unit 3: Control Unit: Instruction types, formats, instruction cycles and subcycles, micro-operations, execution of a complete instruction. Hardwire and micro programmed control: microprogramme sequencing, wide branch addressing, microinstruction with next address field, pre-fetching microinstructions, concept of horizontal and vertical microprogramming.

Notes and shanu kuttan

Unit 4: Memory: Basic concept and hierarchy, semiconductor RAM memories, 2D & 2 1/2D memory organization. ROM memories. Cache memories: concept and design issues 9 performance, address mapping and replacement) Auxiliary memories: magnetic disk, magnetic tape and optical disks Virtual memory: concept implementation.

Notes and shanu kuttan

Unit 5: 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 & asynchronous communication, standard communication interfaces.

Text/Reference Books:

1. M Moris Mano, "Computer System Architecture", Pearson Education, 3rd Ed, 2019.
2. David A. Patterson and John L. Hennessy, "Computer Organization & Design-The Hardware/Software Interface", Morgan Kaufmann, 6th Ed, 2021
3. William Stallings, "Computer Organisation and Architecture, Designing for Performance", Pearson Education Asia, 11th Ed. 2022.
4. Harry F. Jordan and Gita Alaghband, "Fundamentals of Parallel Processing", Pearson Education, 1st Ed. 2003.
5. Barry Wilkinson and Michael Allen, "Parallel Programming", Prentice Hall, 2nd Edition, 2004.
6. Ramesh S. Goankar, "8085 Microprocessors Architecture Application and Programming", Penram International, 6th Edition, 2017.

In csa Watch the Questions of the tutorials only

Course Code	**ITDC0211**
Subject Name	**DISCRETE STRUCTURES**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-0**
Pre-requisites	**NONE**

Course Objectives:

1. To study the objects that, have discrete as opposed to continuous values including the foundations of logic, algorithms and their complexity.
2. To study mathematical reasoning, relations, graphs, trees and combinatorics.

Course Outcomes:

CO1. Explain the basics of discrete mathematics, predicate calculus.
CO2. Understand set theory, relations and functions and recurrence relations.
CO3. Illustrate the concepts of graph theory.
CO4. Experiment with trees to solve problems like minimum spanning tree - traversal of binary tree.
CO5. Explain different algebraic structures.

Course Outcomes	**Program Outcomes**													
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO1 0	PO1 1	PO1 2	PSO 1	PSO2
ITDC0211														
CO 1	M	H	H	H			M							M
CO 2	M	H	M	M	M	M	H							H
CO 3	M		H	M		L					H		M	M
CO 4	M	M				H					M			
CO 5	M		H	M										

Gate Smashers and one shot

Course Content:

Unit 1: INTRODUCTION TO PRELIMINARIES AND PREDICATE CALCULUS: Basic concepts of discrete mathematics and related problems, Propositions and predicates, Disjunction and conjunction, Tautologies and contradiction, Laws of equivalence, Rules of substitution and transitivity, Normal forms, Proof techniques.

One shot

Unit 2: SET THEORY AND FUNCTIONS: Basic concepts, Venn Diagrams, set operations, power set, methods of proof for sets, Relations and ordering, Types of relations, Graph and matrix of a relation, properties of a relation. Functions: definitions and notation, one to one, onto, one to one and onto, composition, Identity and inverse, related results. Counting: Principle of Inclusion and Exclusion, Division and Euclidean Algorithm in Integers, Elements of Probability, Recurrence Relations.

Mohit ppt or Gate smaashers

Unit 3: GRAPH THEORY: Basic concepts of graph theory, multigraphs and weighted graphs, Bipartite graph, walk, path and circuits. Warshall's algorithm: Shortest path, Eulerian paths and circuits, Hamiltonian paths and circuits, Factors of a graph and planar graphs, Graph colorings, Graph isomorphism.

Unit 4: BINARY TREES: Introduction to Trees, complete and extended binary tree, Traversing binary tree, Binary Search Tree, Minimum spanning trees, Heaps, Huffman's algorithm.

Gate Smashers one shot

Unit 5: BASICS OF STRUCTURES: Mathematical induction, Algebraic structures properties, Semi group, Monoid, Group and Sub group - examples and standard results, Generators and evaluation of powers, cosets and Langrange's theorem, Rings, Integral domains, fields.

Text/Reference Books:

1. Tremblay, J. P. and Manohar, R., “Discrete Mathematical structures with applications to Computer Science”, McGraw Hill, 2017.
2. Liu, C.L., “Elements of Discrete Mathematics”, McGraw Hill, 2012.
3. Scheinerman, Edward, "Mathematics: A Discrete Introduction, 3rd Edition", Cengage, 2012.
4. Rosen, Kenneth h., “Discrete Mathematics and Its Applications”, McGraw Hill, 2012.

Course Code	**ITDC0231**
Subject Name	**OBJECT ORIENTED PROGRAMMING LAB**
Contact hours/Credit Scheme (L-T-P-C)	**0-0-2-1**
Pre-requisites	**NONE**

Course Objectives:

1. To make the student learn an object oriented way of solving problems.
2. To teach the student to write programs in C++ to solve the problems.

Course Outcomes:

CO1. Understand the features of C++ supporting object-oriented programming
CO2. Understand the relative merits of C++ as an object-oriented programming language
CO3. Understand how to produce object-oriented software using C++
CO4. Understand how to apply the major object-oriented concepts to implement object-oriented programs in C++, encapsulation, inheritance and polymorphism
CO5. Understand how to use pointer in C++

Course Outcomes	**Program Outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDC0231														
CO 1	H												M	
CO 2	H	L											M	
CO 3		M	H	M	L									M
CO 4		M	M		M							M		M
CO 5	H				M								M	
CO 6	H		M	H	M							M	M	

List of Practicals:

1. Program to Implement Various Control Structures: If statement, Switch case statement, do while loop, For loop, While loop
2. Program to make the use of inline function.
3. Programs to demonstrate the concept of Structure & Unions
4. Programs to implement the concept of Pointer Arithmetic.
5. Program to implement the concept of Functions & Recursion
6. Programs to Understand Different Function Call Mechanism
7. Programs to Understand Storage Specifiers,
8. Program to understand the Use of “this” Pointer.
9. Program to demonstrate the concept of:
    a) Default constructor
    b) Parameterized constructor
    c) Copy constructor
    d) Constructor overloading
10. Program to demonstrate the concept of destructor.
11. Program to show multiple inheritance
12. Program to show multilevel inheritance
13. Program to show hybrid inheritance

14. Program to show the concept of containership.
15. Program to overload unary operator.
16. Program to overload binary operator
17. Program to show the concept of run time polymorphism using virtual function.
18. Program to demonstrate the concept of different kinds of pointers.
19. Program to work with formatted and unformatted IO operations.
20. Program to copy one file onto the end of another, adding line numbers

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents.

Course Code	**ITDC0233**
Subject Name	**DATA STRUCTURE LAB**
Contact hours/Credit Scheme (L-T-P-C)	**0-0-2-1**
Pre-requisites	**NONE**

Course Objectives:

1. To understand how the choice of data structures can lead to efficient implementations of algorithms.
2. To familiarize how certain applications can benefit from the choice of data structures.

Course Outcomes:

CO1. Understand the use of arrays, records, linked structures, stacks, queues, trees, and graphs.
CO2. Understand how the choice of data structures can lead to efficient implementations of algorithms.
CO3. Familiarize how certain applications can benefit from the choice of data structures.
CO4. Apply and implement the learned algorithm for problem solving

Course Outcomes	**Program Outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDC0233														
CO 1	M	L	M			M							M	
CO 2	M	L	M			M						M	M	
CO 3	H	M	M	M		M							M	
CO 4	L		H	M	M	M						H		M

List of Practicals:

1. Implementation of various operations on array like Print the elements of an array in reverse order, delete an element from an array at a given index, find the smallest element in an array, calculate the sum of elements in an array etc.
2. Implementation of linear and binary search technique in an array of n values.
3. Implementation of sparse matrix and perform various operations like search the sparse matrix for an element specified by the user. Print the result of the search appropriately. Use the triple <row, column, value> to represent an element in the sparse matrix.
4. Implementation of various operations on linked list: Creation, Insertion, Deletion, Reverse etc.
5. Implementation of various operations on doubly linked list: Creation, Insertion, Deletion etc.
6. Implementation of stack data structure using array of n values and perform push () and pop () operation.
7. Implementation of stack operations to convert a given infix expression into its equivalent postfix expression. Implement the stack using an array.
8. Implementation of queue data structure using array and linked list with the basic functions of Create(), IsEmpty(), Insert(), Delete() and IsFull().
9. Implementation of circular queue using an array with the following operations: a) Insert a new element b) Delete a given element c) Display the content of queue
10. Implementation of a double ended queue using i) array and ii) doubly linked list respectively.
11. Implementations of Binary Tree menu driven program :
    a. Traverse the tree using all the methods i.e. inorder, preorder & postorder to display the elements in the tree
    b. Copying tree

c. Counting the number of nodes in the tree
d. Counting only leaf nodes in the tree.

12. Implementations of Binary Search Tree (BST) program with at least following operations.
    a) To construct a binary search tree of n values.
    b) To insert new elements in Binary Search Tree (BST)
    c) To delete elements in Binary Search Tree (BST) at various positions
13. Implementation of the following sorting methods to arrange a list of n values in ascending order:
    a) Insertion sort b) Merge sort c) Selection sort d) Bubble sort e) Quick sort f) Heap Sort
14. Implementation of the graph traversal algorithms: a) Depth first traversal b) Breadth first traversal
15. Write a Program to find Longest Common Prefix in a given set of strings (using Trie)
16. Write a Program to find all words matching a pattern in the given dictionary
17. Implementation of Minimal Spanning Trees.

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents.

Course Code	**ITDC0235**
Subject Name	**WEB TECHNOLOGIES LAB**
Contact hours/Credit Scheme (L-T-P-C)	**0-0-2-1**
Pre-requisites	**NONE**

Course Objectives:

1. To understand the best practices in navigation, usability in website design
2. Study of different web technologies.
3. Understand basic principles of web site design, considering the information architecture.

Course Outcomes:

CO1. Understand basic web technologies.
CO2. Identify and incorporate best practices in navigation, usability in website design.
CO3. Learning various scripting and website development.
CO4. Understand and build the skills different web standards and design websites accordingly.

Course Outcomes	**Program Outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDC0235														
CO 1				M	H	M	M		M	H	M	H		
CO 2	M	H	M						M	H			M	
CO 3			H		M	M				L			M	
CO 4	M	M	M			M	M	M			M	M	M	M

List of Practicals:

1. Design a web page using various basic HTML tags.
2. Write a program to display list of items in different styles.
3. Design website using form tag, hyperlinks to other pages. Use frames such that page is divided into 3 frames 20% on left to show contents of pages, 60% in center to show body of page, remaining on right to show remarks. Embed Audio and Video into your HTML web page. Add few form elements such as radio buttons, check boxes and password field. Add a submit button at last.
4. Create a table to show your class time-table. Use tables to provide layout to your HTML page describing your university infrastructure. Use and tags to provide a layout to the above page instead of a table layout.
5. Apply in-line CSS to change colors of certain text portion, bold, underline and italics certain words in your HTML web page. Also change background color of each paragraph using inline CSS.
6. Create a web form for student record. Put validation checks on values entered by the user using JavaScript (such as age should be a value between 1 and 150).
7. Develop a dynamic web pages using JavaScript (client side programming).
8. Write a JavaScript program to display information box as soon as page loads.
9. Implement Servlets:
   (a) create user registration web Application.

(b) At the server end, write code to retrieve contents of request object and show them to the user. Match. user input password with predefined password and show “Valid User” or “Invalid User”.

(c) A web Application name as input and on submit it should show hello <name>. it show start time at the right top corner of the page and provide a logout button. On clicking logout button should goto logout page shows Thank you <name> with duration of usage.

(d) A web application that takes name and age from html page. If age less than 18 it should show Hello <name> You are not authorized to visit the site. Otherwise welcome <name> to website.

10. Create a simple JSP page. Embed JSP in HTML page itself. Separate the JSP and HTML coding in different files and link them together. Add data to request object. At the server end, write code to retrieve contents of request object and show them to the user. Match user input password with predefined password and show “Valid User” or “Invalid User”.
11. Write a program to store the form fields in a database, use any appropriate Server Slide Scripting.
12. Develop a Program using XML. Write a program to connect a XML web page to any database engine.

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents

Course Code	**ITDC0237**
Subject Name	**DATABASE MANAGEMENT SYSTEMS LAB**
Contact hours/Credit Scheme (L-T-P-C)	**0-0-2-1**
Pre-requisites	**NONE**

Course Objectives:

1. To provide a strong formal foundation in database concepts, technology and practice.
2. To make students make become well-informed database application developers.

Course Outcomes:

CO1. To describe data models and schemas in DBMS
CO2. To understand the features of database management systems and Relational database.
CO3. To use SQL- the standard language of relational databases.
CO4. To understand the functional dependencies and design of the database.
CO5. To understand the concept of Transaction and Query processing.

Course Outcomes	**Program Outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDC0237														
CO 1	H													
CO 2	H	M					M						M	
CO 3	H		H		H							M		L
CO 4	H	M		M			M						M	
CO 5	H	M		H			M			M	M	M	M	

List of Practicals:

1. Perform Data Definition Language (DDL) commands in RDBMS
2. Perform Data Manipulation Language (DML) and Data Control Language (DCL)
3. High level language extensions with cursors such as student Grade Calculation and Bill Calculation.
4. Triggers using High level language extension such as Display student result and invalid condition.
5. Procedures and Functions such as insert number and find factorial.
6. Embedded SQL like bonus calculation.
7. Database design using E-R model and Normalization
8. Design and implementation of payroll processing system (Implement and perform)
9. Design and implementation of Banking system
10. Design and implementation of Library Information System
11. Design and implementation of Student Information System
12. Automatic Backup of Files and Recovery of Files

* Students are advised to use **Developer 2000/Oracle-10i** or higher version or other latest version for above listed experiments. Mini Project may also be planned & carried out throughout the semester to understand the important various concepts of Database. *This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents.*

Semester IV

Course Code	**ITDC0202**
Subject Name	**DESIGN AND ANALYSIS OF ALGORITHMS**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives:

1. To develop the understanding of advanced data structures.
2. To develop the understanding of algorithmic design paradigms.

Course Outcomes:

CO 1. Compare, contrast, and apply the key algorithmic design paradigms: brute force, divide and conquer, greedy, dynamic.

CO 2. Compare, contrast, and apply key data structures: trees, lists, stacks, queues, hash tables, and graph representations.

CO 3. Compare, contrast, and apply algorithmic tradeoffs: time vs. space, deterministic vs. randomized, and exact vs. approximate.

CO 4. Implement, empirically compare, and apply fundamental algorithms and data structures to real-world problems.

Course Outcomes	**Program Outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDC0202														
CO 1	M	M	H	M	L								M	
CO 2					M	H							M	
CO 3	L	M		M			L		M				M	
CO 4			M		L						M			H

Course Content:

Unit 1: Introduction: Growth of Functions, analyzing algorithms, Time and space complexity of algorithms, asymptotic notation, standard notations and common functions, The substitution method, The recursion-tree method.

Unit 2: Sorting and order Statistics: Insertion sort, Quick Sort, Merge Sort, Heap Sort, Comparison of sorting algorithms, Sorting in linear time - Counting sort, Radix sort, Bucket sort, medians and order statistics.

Unit 3: Divide and Conquer: Divide and Conquer with examples such as Sorting, Strassen's matrix multiplication; Dynamic Programming: Elements of dynamic programming, Assembly-line scheduling, Matrix-chain multiplication, Longest common subsequence, Optimal binary search trees Greedy

Algorithms: DFS and BFS, Elements of the greedy strategy, Greedy methods with examples such as activity-selection problem, Huffman codes, task-scheduling problem

Unit 4: Minimum Spanning trees: Graph Theory, Growing a minimum spanning tree, Prim's and Kruskal's algorithms. Single source and All-Pairs shortest paths: Dijkstra's and Bellman Ford algorithms, Shortest paths and matrix multiplication, The Floyd-Warshall algorithm, Johnson's algorithm for sparse graphs. Branch and bound with examples such as Travelling Salesman Problem, Graph Coloring, n-queens problem, Hamiltonian cycles. Overview of NP-Completeness: Polynomial time, NP-completeness, NP-complete problems

Unit 5: Advanced Data Structures*:* Red-Black trees, Heaps, Hash tables, Binomial Heaps, Fibonacci Heaps.

Text/Reference Books:

1. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). Introduction to Algorithms (4th ed.). MIT Press.
2. Ullman, J. D., Aho, A. V., & Hopcroft, J. E. (2020). Design and Analysis of Algorithms (2nd ed.). Pearson Education.
3. Kleinberg, J., & Tardos, É. (2021). Algorithm Design (2nd ed.). Pearson Education.
4. Goodrich, M. T., Tamassia, R., & Goldwasser, M. H. (2022). Data Structures and Algorithms in Java (7th ed.). Wiley.

Course Code	**ITDC0204**
Subject Name	**OPERATING SYSTEMS**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives:

1. To understand the services and design of an operating system.
2. To understand the structure and organization of file system
3. To understand the process states and various concepts such as scheduling and synchronization related with it.
4. To understand different memory management approaches.
5. Students should be able to use system calls for managing processes, memory and file system.
6. To understand the data structures and algorithms for implementation of OS.

Course Outcomes:

CO 1. Understand functions, structures and history of operating systems
CO 2. Able to know the design issues associated with operating systems
CO 3. Master various process management concepts such as scheduling, synchronization, multithreading and deadlocks
CO 4. Understand the various concepts associated with memory management such as virtual memory, demand paging, page replacements algorithms
CO 5. Be familiar with various protection and security mechanisms

Course Outcomes	**Program Outcomes**													
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO1 0	PO1 1	PO1 2	PSO 1	PSO 2
ITDC0204													M	
CO 1	M	M	H	M	L									
CO 2					M	H							M	
CO 3	L	M		M										
CO 4								M					M	
CO 5							L							L

Course Content:

Unit 1: Introduction: Operating system functions and characteristics, historical evolution of operating systems, issues in operating system design, Classification of Operating systems- Batch, Interactive, Time-sharing, Real-Time System, Multiprocessor Systems, Multiuser Systems, Multi-process Systems, Multithreaded Systems, Operating System Structure- Layered structure, System Components, Operating System services, Re-entrant Kernels, Monolithic and Microkernel Systems.

Unit 2: Process Management: Process abstraction, process address space, process management, system calls, threads, process hierarchy, Process Control Block (PCB), Process address space, Process identification information, Threads and their management. CPU Scheduling: Scheduling Concepts, Performance Criteria, Process States, Process Transition Diagram, Schedulers, Scheduling Algorithms, Multiprocessor Scheduling, Real time scheduling. Deadlock: System model, Deadlock characterization, Prevention, Avoidance and detection, Recovery from deadlock.

Unit 3: Concurrent Processes: Process Concept, Principle of Concurrency, Producer / Consumer Problem, Mutual Exclusion, Critical Section Problem, Dekker's solution, Peterson's solution, Semaphores, Test and Set operation; Classical Problem in Concurrency- Dining Philosopher Problem, Sleeping Barber Problem; Inter Process Communication models and Schemes, Process generation.

Unit 4: Memory Management: Basic bare machine, Resident monitor, Multiprogramming with fixed partitions, Multiprogramming with variable partitions, Protection schemes, Paging, Segmentation, Paged segmentation, Virtual memory concepts, Demand paging, Performance of demand paging, Page replacement algorithms, Thrashing, Cache memory organization, Locality of reference.

Unit 5: I/O Management and Disk Scheduling: I/O devices, and I/O subsystems, I/O buffering, Disk storage and disk scheduling, RAID. File System: File concept, File organization and access mechanism, File directories, and File sharing, File system implementation issues, File system protection and security.

Text/Reference Books:

1. Tanenbaum, A. S., & Bos, H. (2022). Modern Operating Systems (5th ed.). Pearson Education.
2. Galvin, P. B., Gagne, G., & Silberschatz, A. (2022). Operating System Concepts Essentials (4th ed.). Wiley.
3. Nutt, G. J. (2021). Operating Systems: A Modern Perspective (5th ed.). Addison-Wesley.
4. Stallings, W. (2021). Operating Systems: Internals and Design Principles (10th ed.). Pearson Education.
5. Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). Operating System Concepts (10th ed.). Wiley.
6. Vahalia, U. (2019). UNIX and Linux System Administration Handbook (5th ed.). Prentice Hall.
7. Robbins, K. A., & Robbins, S. (2019). UNIX Systems Programming: Communication, Concurrency and Threads (2nd ed.). Prentice Hall.

Course Code	**ITDC0206**
Subject Name	**SOFTWARE ENGINEERING**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives:

1. Study the current software engineering techniques and examines the software life-cycle, including software specification, design implementation, testing and maintenance.
2. Present software engineering methodologies for the development of Quality, cost-effective, schedule-meeting software.
3. Develop an understanding of ethical and professional issues related to Software Project Delivery.

Course Outcomes:

CO 1. Able to apply the concepts and choose an appropriate SDLC process model for user requirements.
CO 2. To analyze requirement techniques like Data flow diagram, Entity relationship diagram etc.
CO 3. Understanding the concept of Software Design and emphasizing upon various software metrics used for analyzing the software.
CO 4. Demonstrate various testing methodologies and debugging tools for a prototype software.
CO 5. Design various software reliability measures to assess the quality of software in case of various faults and failures.

Course Outcomes	**Program Outcomes**													
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO1 0	PO1 1	PO1 2	PSO 1	PSO 2
ITDC0206														
CO1	M	M				L							M	
CO2		H			L					L			M	
CO3	H		H		M								H	
CO4			L	H	M				M					H
CO5		M	M		L									M

Course Content:

Unit 1: Introduction to Software Engineering: What is Software Engineering? Why Software Engineering, Software Crisis, Notable Changes in Software Development Practices, Software myths. The Software Process: Plan-driven and agile processes, different development philosophies: sequential vs iterative, software development life cycle (SDLC), overview of various SDLC models/methodologies. Plan-Driven Development: The Waterfall Model, Incremental development, Integration and configuration, How to Cope with changes? Software prototyping vs Incremental delivery.

Unit 2: Agile-Software Development: Agility, Extreme Programming, Agile Project Management (Scrum), Scaling out and scaling up, Problem with Agile methods, Combining Agile and plan-driven methods, Agile methods across organizations Process models: The waterfall model, Incremental process models, Evolutionary process models, The Unified process.

Unit 3: Software Requirements: Functional and non-functional requirements, User requirements, System requirements, Interface specification, the software requirements document. Requirements Engineering process: Feasibility studies, Requirements elicitation and analysis, Requirements validation, Requirements management.

Unit 4: Software Design Concepts: Activities carried out during design, Classification of Design Methodologies: Function-Oriented, Object-Oriented, Aspect-oriented, Component-based, Properties of a Good Design, Layered Design, Modularity, Function Oriented Design: Overview of SA/SD Methodology, Data Flow Diagrams (DFDs), DFD Model of a System, DFD Model to Structure Chart, Object-Oriented Design: Objects and object classes, An Object-Oriented design process, UML.

Unit 5: Testing strategies: A strategic approach to software testing, test strategies for conventional software, Black-Box and White-Box testing, Validation testing, system testing, the art of Debugging. Product metrics: Software Quality, Metrics for Analysis Model, Metrics for Design Model, Metrics for source code, Metrics for testing, Metrics for maintenance. Metrics for Process and Products: Software Measurement, Metrics for software quality. Risk management: Reactive vs. Proactive Risk strategies, software risks, Risk identification, Risk projection, Risk refinement, RMMM. Quality Management: Quality concepts, Software quality assurance, Software reliability, The ISO 9000 quality standards. CASE Tools: Types of CASE tools, advantages and components of CASE tools, Unified Modelling Language (UML).

Text/Reference Books:

1. Mall, R. (2021). Fundamentals of Software Engineering (6th ed.). PHI Learning.
2. Sommerville, I. (2022). Software Engineering (11th ed.). Pearson Education.
3. Jawadekar, W. S. (2021). Software Engineering: Principles and Practice (5th ed.). McGraw Hill Education.
4. Pressman, R. S., & Maxim, B. R. (2020). Software Engineering: A Practitioner's Approach (9th ed.). McGraw-Hill Education.

Course Code	**ITDC0208**
Subject Name	**COMPUTER NETWORKS**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives:

1. Understand computer network basic, different models used for study of computer networks, ability to identify different designs, understanding of the issues surrounding wired and wireless Networks.
2. Design, calculate, and apply subnet masks to fulfill networking requirements and building the skills of routing mechanisms.
3. Analyze the features and operations of various application layer protocols such as Http, DNS, SMTP and FTP.
4. Analyze the requirements for a given organizational structure and select the most appropriate networking architecture and technologies
5. Familiarity with the basic protocols of computer networks, and how they can be used to assist in network design and implementation.

Course Outcomes:

CO 1. Build an understanding of the fundamental concepts of computer networking.

CO 2. Familiarize the student with the basic taxonomy and terminology of the computer networking area. Introduce the student to advanced networking concepts, preparing the student for entry into advanced courses in computer networking.

CO 3. Allow the student to gain expertise in some specific areas of networking such as the design and maintenance of individual networks.

CO 4. Understand network security and define various protocols such as FTP, HTTP, Telnet, DNS

Course Outcomes	**Program Outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDC0208														
CO 1	H	M	M	L	H	M	L				M	H	M	
CO 2	H	L	L	L			L	M	M	H	L	H	M	
CO 3	M	M	H					L	M	M	H			M
CO 4	M			M	M					M		M	M	

Course Content:

Unit 1: Introduction: Introduction to Computer Networks History, Network Hardware, Network Software, OSI and TCP/IP reference models.

Unit 2: Physical Layer: Guided Transmission Media, Wireless Transmission, Communication Satellites, The Public Switched Telephone Network.

Unit 3: Data Link Layer: Data Link Layer Design Issues, Error Detection and Correction, Elementary Data Link Protocols, Sliding Window Protocols, Example Data Link Protocols. Medium Access Control Layer: The Channel Allocation Problem, Multiple Access Protocols, Ethernet, Wireless Lans, Broadband Wireless, Bluetooth, RFID, Data Link Layer Switching.

Unit 4: Network Layer: Network Layer Design Issues, IP, Routing Algorithms, Congestions Control Algorithms, Quality of Service, Routing Information Protocol

Unit 5: Transport Layer: The Transport Service, Elements of Transport Protocols, Flow and Congestion Control, The Internet Transport Protocol: UDP, TCP.

Unit 6: Application Layer: Hyper Text Transfer Protocol, File Transfer Protocols and Trivial File Transfer Protocol, Simple Mail Transfer Protocol, Domain Name Service, Telent, Secure shell, Network File System, Simple Network Management Protocol, Dynamic Host Configuration Protocol, Multipurpose Internet Mail Extensions, Post Office Protocol.

Text/Reference Books:

1. Tanenbaum, A. S., & Wetherall, D. J. (2022). Computer Networks (6th ed.). Pearson Education.
2. Kurose, J. F., & Ross, K. W. (2022). Computer Networking: A Top-Down Approach (8th ed.). Pearson Education.
3. Stallings, W. (2021). Data and Computer Communications (11th ed.). Pearson Education.
4. Forouzan, B. A., & Mosharraf, F. (2022). Data Communications and Networking (6th ed.). McGraw-Hill Education.
5. Peterson, L. L., & Davie, B. S. (2023). Computer Networks: A Systems Approach (6th ed.). Morgan Kaufmann.

Course Code	**ITDC0210**
Subject Name	**FORMAL LANGUAGE AND AUTOMATA THEORY**
Contact hours/Credit Scheme (L-T-P-C)	**3-1-0-4**
Pre-requisites	**NONE**

Course Objectives:

1. Demonstrate advanced knowledge of formal computation and its relationship to languages.
2. Distinguish different computing languages and classify their respective types.
3. Recognize and comprehend formal reasoning about languages.

Course Outcomes:

CO 1. Understand theoretical foundations of computer science.
CO 2. Master regular languages, finite automata, pushdown automata, Turing recognizable Languages.
CO 3. Employ finite stated machines to solve problems in computing.
CO 4. Think analytically and intuitively for problem-solving situations in related areas of theory in computer science.

Course Outcomes	**Program Outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDC0210														
CO 1	H	M	M	L								L	H	H
CO 2	L			H	H								M	L
CO 3						M		M					H	
CO 4					L					H	H		H	H

Course Content:

Unit 1: Basics in Theory of Computations: Basic concepts of strings, alphabets, languages, Principles of Mathematical Induction.

Finite Automata: Automata and Applications of Automata Theory, Deterministic and Non-Deterministic FA, Comparison and Equivalence of DFA and NFA.

Finite State Machines: Moore and Mealy Machines, Equivalence of Moore and Mealy Machines.

Regular Expressions: Regular Expression, Equivalence of Regular Expression and Finite Automata, Equivalence of Regular Grammar, and Finite Automata, Regular and Non- Regular Languages, Pumping Lemma for Regular Sets.

Unit 2: Context Free Language: Context Free Grammar, Derivation trees, Context Free Grammar Simplification, Chomsky & Greibach Normal forms, Ambiguities.

Languages and Grammars: Construct of a language, Grammar, Chomsky Classification of Formal Languages.

Pushdown Automata: Definition, Equivalence of PDA by Empty Store and PDA by Final State. Construction of PDA for CFLs.

Unit 3: Turing Machines: Introduction and Turing Machine Model, Computable functions and languages. Techniques for construction of Turing machines, Church's Hypothesis.

Unit 4: Undecidability: Recursive and recursively enumerable languages, Rice theorem, Post's correspondence problem.

Text/Reference Books:

1. Hopcroft, J. E., Motwani, R., & Ullman, J. D. (2017). Introduction to Automata Theory, Languages, and Computation (3rd ed.). Pearson Education.
2. Mishra, K. L. P., & Chandrasekaran, N. (2022). Theory of Computer Science: Automata, Languages and Computation (4th ed.). PHI Learning.
3. Lewis, H. R., & Papadimitriou, C. H. (2018). Elements of the Theory of Computation (2nd ed.). Pearson Education.
4. Linz, P. (2017). An Introduction to Formal Languages and Automata (6th ed.). Jones & Bartlett Learning.
5. Sipser, M. (2019). Introduction to the Theory of Computation (4th ed.). Cengage Learning.

Course Code	**ITFE0202**
Subject Name	**COMPETITIVE PROGRAMMING**
Contact hours/Credit Scheme (L-T-P-C)	**1-0-0-1**
Pre-requisites	**NONE**

Course Objectives: The focus of the course is the

1. Learn to select appropriate algorithms for a given problem, integrate multiple algorithms
2. Solving a complex problem, design new algorithms, and implement them in C or C++.
3. Solving problems in teams and work under time pressure.
4. Students will learn to break down complex problems into manageable components and devise efficient algorithms and strategies to solve them.

Course Outcomes:

CO1. Be familiar with standard competitive programming strategies and effective team collaboration techniques

CO2. To implement efficient solutions to programming problems while working under time pressure

CO3. To recognize the appropriateness and application of standard algorithmic strategies to new and challenging problems.

CO4. Apply the String and mathematical algorithms for problem solving.

Course Outcomes	**Program Outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITFE0202														
CO 1.	H	M	L	M	L				H	M				
CO 2.	M	M		L						M	L			M
CO 3.	M	H		L	M								M	
CO 4.	H		M	H	M									

Course Content:

Unit 1: Introduction: Introduction to Competitive Programming, What is Competitive Programming, Calculating Time Complexity of Code, Growth of Functions, analyzing algorithms, Time Complexity in Competitive Programming, asymptotic notation, standard notations and common functions, The substitution method, The recursion-tree method.

Unit 2: Bit manipulation: Importance of Bit Manipulation Techniques, Basic Binary Operations, Binary Exponent, Bit Operations, Check i^{th} bit is SET or NOT SET, Toggle ith bit, Check odd or even using bits, Check if Number is Power of 2, Need for Modulo 10^9 + 7, Modulo Arithmetic. Strings and Pointers.

Unit 3: Tree and Graphs: Tree Traversal algorithms: Inorder, Preorder and Postorder, Binary Search Trees (BST), Insertion and Deletion in BST, Complexity of Search Algorithm, Graph: Adjacency Matrices, Adjacency List, Graph Traversal: Depth First Search and Breadth First Search, Spanning Trees, Minimum Cost Spanning Trees: Prims and Kruskal algorithm. Transitive Closure and Shortest Path algorithm: Warshall Algorithm and Dijkstra Algorithm, Heap: Introduction and basic operations on heaps.

Unit 4: Hashing: Hash Function, Collision Resolution Strategies Storage Management: Garbage Collection and Compaction.

Text/Reference Books:

1. Guide to Competitive Programming: Learning and Improving Algorithms Through Contests by Antti Laaksonen (2nd Edition, 2021)
2. Competitive Programming 4: The New Lower Bound of Programming Contests by Steven Halim and Felix Halim (2021)
3. Competitive Programmer's Handbook by Antti Laaksonen (2021)
4. Python Competitive Programming: Mastering Algorithms with Examples by Ranjan Vivek (2022)
5. Competitive Programming in Python: 128 Algorithms to Develop Your Coding Skills by Dami Lee (2022)

Course Code	**ITDC0232**
Subject Name	**DESIGN AND ANALYSIS OF ALGORITHMS LAB**
Contact hours/Credit Scheme (L-T-P-C)	**0-0-2-1**
Pre-requisites	**NONE**

Course Objectives:

1. To introduce mathematical aspects and analysis of algorithms.
2. To introduce algorithm design methods

Course Outcomes:

CO1. Identify the problem given and design the algorithm using various algorithm design techniques.
CO2. Implement various algorithms in a high-level language.
CO3. Analyze the performance of various algorithms.
CO4. Compare the performance of different algorithms for same problem

Course Outcomes	**Program Outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDC0232														
CO 1	M	M	H	L		L			L	M			H	
CO 2	M		M		L					M	L			H
CO 3	M	H			L								M	
CO 4		M	L			L				L				M

List of Practicals:

1. Implementation of Quick sort using the Divide and Conquer technique and analyze its Time Complexity.

2. Implementation of Merge sort using the Divide and Conquer technique and analyze its Time Complexity.
3. Implementation of Radix sort and analyze its Time Complexity.
4. Implementation of Counting sort and analyze its Time Complexity.
5. Implementation of Bucket sort and analyze its Time Complexity.
6. Implementation of Hash table and supported operations.
7. Implementation of red black trees and supported operations.
8. Implementation of Max/Min Heap and supported operations
9. Implementation of the Minimum Spanning tree using prim's algorithm and kruskal's algorithm.
10. Implementation of matrix multiplication problem using the dynamic programming approach.
11. Implementation of assembly line scheduling using the dynamic programming approach.
12. Implementation of the longest common subsequence problem using the dynamic programming approach.
13. Implementation of the Optimal Binary Search Tee problem using the dynamic programming approach.
14. Implementation of the knapsack problem using greedy method.
15. Implementation to find the solution to the N queen's problem using backtracking.
16. Implementation to find the shortest path using Floyd's algorithm.
17. Implementation to solve Graph Coloring problem and Hamiltonian Cycle Problem.

This is only the suggested list of Practicals. Instructor may frame the Practicals relevant to the course contents.

Course Code	**ITDC0234**
Subject Name	**OPERATING SYSTEMS LAB**
Contact hours/Credit Scheme (L-T-P-C)	**0-0-2-1**
Pre-requisites	**NONE**

Course Objectives:

1. Analyse the working of an operating system and its components and Define and analyse the synchronization process.
2. Identify the working methodology of multithreaded applications.
3. Compare and analyse different file systems being used in different operating systems.
4. Design and implement system-level applications for open-source operating systems

Course Outcomes:

CO1. Analyse the working of an operating system and its components.
CO2. Define and analyse the synchronization process.
CO3. Identify the working methodology of multithreaded applications.
CO4. Compare and analyse different file systems being used in different operating systems.

Course Outcomes	**Program Outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDC0234														
CO 1	M	L		H		L				M			M	
CO 2	M	M	H										M	
CO 3	M	L		M			M			M				M

CO 4	L	M	H	M						L	L			H

List of Practicals:

1. Simulation of the CPU scheduling algorithms
    a) Round Robin
    b) Shortest Job First (SJF)
    c) First Come First Serve (FCFS)
    d) Priority
2. Simulation of MUTEX and SEMAPHORES.
3. Demonstration of various thread related concepts.
4. Simulation of Bankers Deadlock Avoidance and Prevention algorithms.
5. Implementation of Process Synchronization (Reader-Writer, Sleeping Barber and Dining Philosopher's Problem)
6. Simulation of Page Replacement Algorithms
    a) First In First Out (FIFO)
    b) Least Recently Used (LRU)
    c) Least Frequently Used (LFU)
7. Simulation of paging techniques of memory management.
8. Simulation of file allocation Strategies
    a) Sequential
    b) Indexed
    c) Linked
9. Simulation of file organization techniques
    a) Single Level Directory
    b) Two Level
    c) Hierarchical
    d) Directed Acyclic Graph (DAG)
10. Simulation of the Disk scheduling algorithms
    a) FCFS
    b) SCAN)
    c) C-SCAN
11. Demonstration of the Shell Programming.

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents.

Course Code	**ITDC0236**
Subject Name	**SOFTWARE ENGINEERING LAB**
Contact hours/Credit Scheme (L-T-P-C)	**0-0-2-1**
Pre-requisites	**NONE**

Course Objectives:

1. To understand the software engineering methodologies involved in the phases for project development.
2. To gain knowledge about open source tools used for implementing software engineering methods.
3. To exercise developing product-startups implementing software engineering methods.
4. Open source Tools: StarUML / UMLGraph / Topcased

Course Outcomes:

CO1. Able to prepare SRS, design, test cases, software configuration management and risk management related document.

CO2. Develop function oriented and object oriented software design using tools like rational rose

CO3. Implement unit testing and integration testing.

CO4. Able to track the progress of a project using OpenProject or equivalent tool.

Course Outcomes	Program Outcomes													
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12	PSO 1	PSO 2
ITDC0236														
CO1		H		L						M	M		M	M
CO2	H		H		M									M
CO3			L	H		M		L	M		H			H
CO4			M	M	H		H		H	M	H	M	M	M

List of Practicals:

1. Introduction to Project Management Software.
2. Software process overview.
3. Project planning.
4. Software Requirements Specification (SRS).
5. Introduction to Unified Modeling Language (UML) and use case diagrams.
6. Flow of events and activity diagram.
7. Object Oriented Analysis: discovering classes and class diagrams.
8. Interaction diagrams: sequence and collaboration diagrams.
9. Software Design: software architecture and object-oriented design.
10. Effort and cost estimation using Constructive Cost Model (COCOMO)
11. State Transition Diagram.
12. Component and deployment diagrams.
13. Issue Tracking Systems like GitHub, BitBucket.
14. Software Configuration Management tools like CVS or SVN.
15. Software testing.

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents.

Course Code	**ITDC0238**
Subject Name	**COMPUTER NETWORKS LAB**
Contact hours/Credit Scheme (L-T-P-C)	**0-0-2-1**
Pre-requisites	**NONE**

Course Objectives:

1. To understand the working of the fundamental elements of communications systems.
2. Study of different modulation techniques.

Course Outcomes:

CO1. Understand basic computer network technology, data communications System and its components.

CO2. Identify the different types of network topologies and protocols, to enumerate the layers of the

OSI model and TCP/IP

CO3. Identify the different types of network devices and their functions within a network.

CO4. Understand and build the skills of subnetting and be familiar with the basic protocols of computer networks, and how they can be used to assist in network design and implementation.

Course Outcomes	**Program Outcomes**													
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12	PSO 1	PSO2
ITDC0238														
CO 1	H	M	M	L	H	M	L				M	H	M	
CO 2	H	L	L	L			L	M	M		L	H	M	
CO 3	M	M	H					L	M	M	H			M
CO 4	M			M	M					M		M	M	

List of Practicals:

1. Network Physical Components Hands-on (Networks Cabling)
2. Brief introduction to Cisco Packet Tracer covering network devices, cables and end-devices. Connect one PC to another PC using the cable and also design a network of few computers using hub/switch and assign IP address and subnet mask to them. Implement Star topologies in Packet Tracer.
3. Implement Bus, Ring, Mesh, and Hybrid topology. Assign IP address and subnet mask to each computer and run the ping command to check the reachability of the systems. Send message between source and destination and observe the flow of the messages.
4. Design two separate network and connect them using Router
5. Implementation and performance of STOP and Wait protocol
6. Implementation and performance of Sliding Window protocol
7. Design Static route configuration, Dynamic route configuration, and Default route configuration
8. Design Network Address Translation and Port Address Translation
9. Design virtual local area network (VLAN)
10. Experiencing Real-world Network Infrastructure
11. Experiencing Real-world network Devices using Switches and Routers
12. Implementing Client/Server Applications
13. Packet Capturing and Analysis using the monitoring tool
14. Application test using the network monitoring tool

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents

Semester-V

Course Code	**ITDC0301**
Subject Name	**OBJECT-ORIENTED MODELING AND DESIGN USING UML**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives:

1. Understanding basic principles modeling of software systems using UML 2.0.
2. Theoretical and practical preparation enabling students to work in project teams.

Course Outcomes:

CO1. Explain the object- oriented software development process, including object-oriented methodologies and work flow.
CO2. Collect requirements and prepare their scenarios
CO3. Justify designs based on design principles, patterns, and heuristics.
CO4. Prepare different UML diagrams.

Course Outcomes	**Program Outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDC0301														
CO 1	H	M	M	L										
CO 2	L			H	H							L		M
CO 3						M		M			L			M
CO 4					L					H		L		M

Course Contents:

Unit 1: Introduction: What is Object Orientation? What is OO development? OO themes, OO modeling history, Modeling as Design Technique: Modeling, abstraction, The three models. Process Overview: Development stages, Development life cycle.

Unit 2: Project Organization Concepts: Project Organizations, Roles, Tasks and Work Products, Schedule. Project Communication Concepts: Planned Communication, Unplanned Communication Communication Mechanisms, Organizational Activities

Unit 3: Analysis and System Conception: Problem Analysis, Problem Domain Classes, Defining the problem and the scope, Requirements Engineering, Types of Requirements. Requirements Validation, Completeness, Consistency, Clarity, and Correctness, Realism, Verifiability, and Traceability, Greenfield Engineering, Reengineering, and Interface Engineering. Devising a system concept, elaborating a concept, preparing a problem statement, domain Analysis: Overview of analysis.

Unit 4: Modeling: Class Modeling: Object and class concepts, Link and associations concepts, Generalization and inheritance, A sample class model, Navigation of class models, Advanced object and class concepts, Aggregation, Abstract classes, Packages. State Modeling: Events, States, Transitions and Conditions, State diagrams, State diagram behavior, relation of class and state models. Interaction Modeling: Use case models, Sequence models, Activity models. Use case relationships.

Unit 5: Patterns: What is a pattern and what makes a pattern? Properties of Patterns, Pattern – A Three-part Schema, Different Types of Patterns: Layer Pattern, Broker Pattern, Shared-Data Pattern, Pipe and Filter Pattern, Model-View-Controller Pattern, Client-Server Pattern, Peer-to-Peer Pattern, Microkernel Pattern, Blackboard, Relationship between tactics and patterns, using tactics together.

Text/Reference Books:

1. Bernd Bruegge & Allen H. Dutoit, "Object-Oriented Software Engineering Using UML, Patterns, and Java" Third Edition, Pearson, 2010.
2. Len Bass, Paul Clements, Rick Kazman, Software Architecture in Practice, Pearson Education, Third edition, 2013.
3. Michael Blaha, James Rumbaugh, "Object Oriented Modeling and Design with UML", Second Edition, PHI, 2012.
4. Meilir Page-Jones, "Fundamentals of Object-Oriented Design in UML", Pearson Education (2008).
5. Simon Benett, Steve Mc Robb, "Object Oriented Systems Analysis and Design using UML", Second Edition, TMH (2007).

Course Code	**ITDC0303**
Subject Name	**JAVA PROGRAMMING**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**None**

Course Objectives:

1. Building robust applications using Java's object-oriented features.
2. Understanding the usage of java class libraries.
3. Building multithreaded, platform-independent and GUI based java applications for business problems.

Course Outcomes:

CO 1. Knowledge of the structure and model of the Java programming language
CO 2. Write Java programs that solve simple business problems.
CO 3. Create java applications that are robust and multithreaded.
CO 4. Write simple GUI interfaces for a program to interact with users, and to understand the event-based GUI handling principles.

Course Outcomes	**Program Outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDC0303														
CO 1	M				L							L		M
CO 2		H						M						
CO 3			M			H							L	
CO 4		H			H				L					

Course Content:

Unit-1: Introduction to java programming, Features of java programming, Review of basic OOP Concepts in Java, this keyword, super keyword, garbage collection, Access control: public, private, protected and default. Packages and Interfaces: Defining, Creating and Accessing a Package, Understanding CLASSPATH, importing packages, differences between classes and interfaces, defining an interface, implementing interface, applying interfaces, variables in interface and extending interfaces. Exploring packages – Java.io, Java.util.

Unit 2: **Exception handling and multithreading:** Concepts of exception handling, benefits of exception handling, Termination or resumptive models, exception hierarchy, usage of try, catch, throw, throws and finally, built in exceptions, creating own exception subclasses.

Differences between multithreading and multitasking, thread life cycle, creating threads, synchronizing threads, daemon threads, thread groups.

Unit 3: JDBC: Database Programming using JDBC, JDBC Driver Types, JDBC Packages, Overview of the JDBC Process, Database Connection, statement Objects, Result set, Transaction Processing, Metadata, Data Types, Exceptions.

Unit 4: Event Handling: Events, Event sources, Event classes, Event Listeners, Delegation event model, handling mouse and keyboard events, Adapter classes, inner classes. The AWT class hierarchy, User interface components (labels, button, canvas, scrollbars, text components, check box etc.) **Swing:** Introduction, limitations of AWT, MVC architecture, components, containers, exploring swing, JApplet, JFrame and JComponent, Icons and Labels, text fields, buttons – The JButton class, Check boxes, Radio buttons, Combo boxes, Tabbed Panes, Scroll Panes, Trees, and Tables.

Unit 5: Networking: Basics of network programming, addresses, ports, sockets, simple client server program, multiple clients, Java .net package, java.util packages,

Text/Reference Books:

1. Marc Loy, Patrick Niemeyer, and Daniel Leuck, "Learning Java", 4th Ed. 2024, O'Reilly Media
2. Herbert Schildt, "Java: The Complete Reference", 12th Edition, 2022, McGraw Hill
3. Cay S. Horstmann, "Core Java Volume I – Fundamentals", 12th EdPearson, 2021.
4. Joshua Bloch, "Effective Java", 3rd Edition, Addison-Wesley Professional, 2018
5. Y. Daniel Liang, "Introduction to Java programming", 7th Ed., Prentice Hall, 2008.
6. R.A. Johnson-Thomson , "An introduction to Java programming and object oriented application development", 1st Ed., Cengage Learning, 2006.

Course Code	**ITDC0305**
Subject Name	**ARTIFICIAL INTELLIGENCE**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives:

1. To provide a strong foundation of fundamental concepts in Artificial Intelligence
2. To provide a basic exposition to the goals and methods of Artificial Intelligence
3. To enable the student to apply these techniques in applications which involve perception, reasoning and learning

Course Outcomes:

CO 1. Identify problems that are amenable to solution by AI methods, and which AI methods may be suited to solving a given problem.
CO 2. Formalize a given problem in the language/framework of different AI methods
CO 3. Describe basic AI algorithms (e.g., standard search algorithms or resolution).
CO 4. Design and carry out an empirical evaluation of different algorithms on a problem formalization.

Course Outcomes	**Program Outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDC0305														
CO 1	H	M				L		L						
CO 2			M	H	H								L	
CO 3		M		M	L				M					
CO 4	H				H							H		

Course Content:

Unit 1: Introduction to AI: Definitions, Historical foundations, Basic Elements of AI, Characteristics of intelligent algorithm, AI application Areas, Constraints and Optimization

Unit 2: Problem solving: State space search; Production systems, search space control: depth-first, breadth-first search, Heuristic functions, heuristic search - A*, AO*, Hill climbing, best-first search, branch and bound. Problem Reduction, Constraint Satisfaction End, Means-End Analysis, Min-Max, alpha beta pruning.

Unit 3: Knowledge Based Systems: Proportional Logic, FOPL, Clausal Form, Resolution & Unification. Knowledge representation, acquisition, organization & Manipulation, Semantic nets,

Frames, Conceptual Dependency, Scripts & CYC. Handling uncertainty: Non-Monotonic Reasoning, Probabilistic reasoning, use of certainty factors, Fuzzy logic.

Unit 4: Machine Learning: Concept of learning, Concept creation, learning automation, supervised and Unsupervised Learning, Expert Systems: Need and justification for expert systems, Basic Components & architecture of Expert systems, ES-Shells, Representing & Using Domain Knowledge, Knowledge acquisition in expert Systems. Case studies: MYCIN, RI.

Text/Reference Books:

1. E. Rich, K. Knight and S.B. Nair, "Artificial Intelligence", Third Edition, McGraw Hill Education, 2017.
2. George F. Luger, "Artificial Intelligence – Structures and Strategies for Complex Problem Solving", Sixth Edition, Pearson Education, 2021.
3. Stuart Russell, Peter Norvig, "Artificial intelligence: A Modern Approach", Fourth edition, Prentice Hall, 2022.
4. Dan W. Patterson, "Introduction to Artificial Intelligence and Expert Systems", First Edition, Pearson Education India, 2015.
5. Nils J. Nilson, "Principles of Artificial Intelligence", Narosa Publishing Co., First Edition, 1993.

Course Code	**ITDC0307**
Subject Name	**CRYPTOGRAPHY AND NETWORK SECURITY**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives:

1. To understand mathematical foundations of cryptography theory.
2. To comprehend entity authentication and key management.
3. To describe network layer security and analyze network security protocols.
4. To understand transport layer security protocols in Internet.
5. To introduce application layer security.
6. To develop security protocols and cryptosystem for real time applications.

Course Outcomes:

CO 1. To understand mathematical foundations of cryptography theory.
CO 2. To establish the entity authentication and key management.
CO 3. To analyze network layer and transport layer security.
CO 4. To develop security protocols and cryptosystem for real time applications.

Course Outcomes	**Program Outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDC0307														
CO 1	M		M			H		M		M		L	L	
CO 2	M	H	M	M		H		H				H	M	
CO 3			H	M	H	M		H	H			M		M
CO 4				M	M				M		M			

Course Content:

Unit 1: Introduction to security attacks, services and mechanism, classical encryption techniques- substitution ciphers and transposition ciphers, steganography, stream and block ciphers, modern block ciphers: block ciphers principles, Shannon's theory of confusion and diffusion, fiestal ciphers structures. cryptanalysis

Unit 2: Introduction to group, field, finite field of the form GF(p), modular arithmetic, prime and relative prime numbers, Extended Euclidean Algorithm, Data Encryption Standard (DES), strength of DES, idea of differential cryptanalysis, block cipher modes of operations, Triple DES, Advanced Encryption Standard (AES) encryption and decryption
Fermat's and Euler's theorem, primality testing, Chinese remainder theorem, computational hard problems: integer factorization, discrete logarithmic problem, principles of public key crypto systems, RSA algorithm, security of RSA

Unit 3: Message authentication codes: authentication requirements, authentication functions, message authentication code, hash functions, birthday attacks, security of hash functions, secure Hash Algorithm (SHA)

Digital signatures: digital signatures, elgamal digital signature techniques, Digital Signature Standards (DSS), proof of digital signature algorithm.

Key management and distribution: Symmetric key distribution, Diffie -Hellman key Exchange, Public key distribution, X.509 Certificates, Public key Infrastructure.

Unit 4: Authentication applications: Password Based-Authentication, Challenge Response-based Authentication, Kerberos

IP security: architecture, authentication header, encapsulating security payloads, combining security associations, key management. Introduction to Secure Socket Layer, Transport layer Security. Electronic mail security: Pretty Good Privacy (PGP), S/MIME.

Text/Reference Books:

1. William Stallings, "Cryptography and network Security", Pearson Education 2003.
2. Trappe & Washington, "Introduction to Cryptography with Coding Theory", Prentice-Hall 2001.
3. D Stinson, "Cryptography: Theory and Practice", Second Edition Chapman & Hall 2002.
4. Kaufman, Perlman, and Speciner, "Network Security", Prentice-Hall Second Edition 2001.
5. Michael E. Whitman, "Principles of information Security" , Cengage learning, New Delhi

Course Code	**ITDC0333**
Subject Name	**JAVA PROGRAMMING LAB**
Contact hours/Credit Scheme (L-T-P-C)	**0-0-2-1**
Pre-requisites	**NONE**

Course Objectives:

1. To make the student learn the application of advanced object oriented concepts for solving problems.
2. To teach the student to write programs using advanced Java features to solve the problems

Course Outcomes:

CO 1. Knowledge of the structure and model of the Java programming language
CO 2. Analyze the Java programming language for various programming technologies
CO 3. Evaluate user requirements for software functionality required to decide whether the Java programming language can meet user requirements
CO 4. Use Java Script to validate user input and create good, effective and customized websites

Course Outcomes	**Program Outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDC0333														
CO 1	M				L							L		M
CO 2		H						M						
CO 3			M			H							L	
CO 4		H			H				L					

List of Practicals:

1. Program to find total, average of given two numbers by using function with default arguments, static data members and this keyword.
2. Program to illustrate class, objects and constructors.

3. Program to create a class complex with necessary operator overloading and type conversion such as integer to complex, complex to double.
4. Program that randomly generates complex numbers and write two numbers per line in a file along with an operator(+,-,P,*,/) .The numbers are written to file in the format (a+ib)
5. Program to read one line at a time, perform the corresponding operation on two
6. complex numbers read, write the result to another file (one per line)
7. Program to illustrate inheritance (Student Evaluation)
8. Program to handle the situation of exception handling.
9. Program to demonstrate the concept of polymorphism.
10. Program to illustrate Method Overriding?
11. Program to illustrate Method overloading of assignment operator?
12. Program to illustrate Array Manipulation?
13. Program to illustrate Synchronization?
14. Program to String Tokenizer?
15. Program to implement the concept of User defined Exceptions.
16. Program to illustrate the use of creation of packages.
17. Program to illustrate Multithreading and Multitasking?
18. Program to illustrate thread priorities.
19. Implement a Java Servelet for HTTP Proxy Server.
20. Use JSP pages for sharing session and application data of HTTP Server.
21. Program to use JDBC connectivity program for maintaining database by sending queries.

This is only the suggested list of Practicals. Instructors may frame additional Practicals relevant to the course contents.

Course Code	**ITDC0335**
Subject Name	**ARTIFICIAL INTELLIGENCE LAB**
Contact hours/Credit Scheme (L-T-P-C)	**0-0-2-1**
Pre-requisites	**NONE**

Course Objectives:

1. To make the student learn the application of advanced object-oriented concepts for solving problems.
2. To teach the student to write programs using advanced Java features to solve the problems

Course Outcomes:

CO 1. Knowledge of the structure and model of the Java programming language.
CO 2. Use the Java programming language for various programming technologies.
CO 3. Evaluate user requirements for software functionality required to decide whether the Java programming language can meet user requirements.

Course Outcomes	**Program Outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDC0335														
CO 1	H	M				L		L						M
CO 2			M	H	H								L	
CO 3		M		M	L				M					
CO 4	H				H							H		

List of Practicals:

1. Implement Uninformed Search Strategies: BFS, DFS.
2. Implement ID (Iterative Deepening- Idea of Game Searching (Chess, Checkers, Tic-Tac-Toe, Puzzle Game etc.))
3. Implement Informed (heuristic) search strategy for optimal shortest path and traffic navigational system. (Hint: A* algorithm by mentioning OPEN and CLOSED List and it takes a heuristic function as Euclidian Distance Equation)
4. Implement Informed (heuristic) search strategy for problem reduction and for planning good algorithms. (Hint: AO*-AND/OR graph by mentioning OPEN and CLOSED List).
5. Understand Text Summarization and Implement your own summarizer in python.
6. Implement Bayes rule with probabilistic inference for reasoning under uncertainty.
7. Case Study1: Natural language processing and Expert System.
8. Case Study2: Briefly mention any three artificial intelligence libraries in Python
9. Write a python program to implement linear regression and plot the graph.
10. Performance analysis on a specific dataset (Mini project).

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents.

Course Code	**ITDC0337**
Subject Name	**CRYPTOGRAPHY AND NETWORK SECURITY LAB**
Contact hours/Credit Scheme (L-T-P-C)	**0-0-2-1**
Pre-requisites	**NONE**

Course Objectives:

1. Understand cryptography and network security concepts and application
2. Identify and investigate network security threat
3. Analyze and design key management protocols
4. Analyze and design network security protocols

Course Outcomes:

CO 1. Implement entity authentication protocols
CO 2. Exchange secret keys using key management protocols
CO 3. Design and implement cryptographic primitives

Course Outcomes	**Program Outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDC0337														
CO 1	M		M			H		M		M		L	L	
CO 2	M	H	M	M		H		H				H	M	
CO 3			H	M	H	M		H	H			M		M
CO 4				M	M				M		M			

List of Practicals:

1. Implement the encryption and decryption of 8-bit data using ‘Simplified DES Algorithm’ in ‘C’.
2. Implement ‘Linear Congruential Algorithm’ to generate 5 pseudo-random numbers in ‘C/Python’.
3. Implement Needham-Shroeder key establishment protocol.
4. Implement Otway Rees key establishment protocol.
5. Implement Diffie-Hellman Key Exchange Algorithm in ‘C/Python’.
6. Implement Simple Password Based-Entity Authentication in ‘C/Python’.
7. Implement variants of Password Based-Entity Authentication in ‘C/Python’.
8. Implement Challenge-Response Based Entity Authentication in ‘C/Python’.
9. Implement RSA algorithm for encryption and decryption in ‘C/Python’.
10. Configure a mail agent to support Digital Certificates, send a mail and verify the correctness of this system using the configured parameters.
11. Configure SSH (Secure Shell) and send/receive a file on this connection to verify the correctness of this system using the configured parameters.

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents.

Semester VI

Course Code	**ITDC0302**
Subject Name	**DISTRIBUTED SYSTEM AND CLOUD COMPUTING**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives:

1. To study how to use Cloud Services.
2. To implement Virtualization
3. To implement Task Scheduling algorithms.
4. Apply Map-Reduce concept to applications.

Course Outcomes:

CO 1. To understand and apply the knowledge of systems protocols and mechanisms to support Cloud computing.

CO 2. To analyze and implement the concepts of Cloud Computing.

CO 3. To deploy applications on cloud middleware.

CO 4. To understand and resolve security issues in various applications.

Course Outcomes	**Program Outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDC0302														
CO 1	L	H		M									L	
CO 2	L	M	M					M						L
CO 3	L		M			H								
CO 4	L	M	M		M									

Course Content:

Unit 1: Evolution of Distributed System, characteristics, design and challenges, Processes and Threads, Physical and logical clocks, global states, mutual exclusion, election algorithms, Cloud Computing, History of Cloud Computing, Architecture, Applications, deployment models, and service models, Cloud Storage, Why Cloud Computing Matters, Advantages of Cloud Computing, Disadvantages of Cloud Computing, Companies in the Cloud Today.

Unit 2: Virtualization: Issues with virtualization, virtualization technologies and architectures, Internals of virtual machine monitors/hypervisors, virtualization of data centres, and Issues with Multi-tenancy. Implementation: Study of Cloud computing Systems like Amazon EC2 and S3, Google App Engine, and Microsoft Azure, Build Private/Hybrid Cloud using open-source tools, Deployment of Web Services from Inside and Outside a Cloud Architecture, Dynamic Scalability Architecture.

Unit 3: Resource Management: Server consolidation, Dynamic resource provisioning, Resource Optimization, Scheduling Techniques for Advance Reservation, SLA Requirements, and Load Balancing, SLA Management, various load balancing techniques Broad Aspects of Migration into

Cloud, Migration of virtual Machines and techniques. Fault Tolerance Mechanisms.

Unit 4: Security: Vulnerability Issues and Security Threats, Application-level Security, Data level Security, and Virtual Machine level Security, Infrastructure Security, and Multi-tenancy Issues. IDS: host-based and network-based, Security-as-a-Service. Trust Management, Identity Management, and Access Controls Techniques.

Unit 5: Advances: Fog Computing, Edge Computing, Green Cloud, Cloud of Things, Mobile Cloud Computing, Serverless Computing, Web of Things.

Text/Reference Books:

1. Sunil Kumar Manvi, Gopal Shyam, Cloud Computing, Concepts and Technologies, CRC press 2021.
2. Haley Beard, Cloud Computing Best Practices for Managing and Measuring Processes for On-demand Computing, Applications and Data Centers in the Cloud with SLAs, Emereo Pty Limited, July 2008.
3. Thomas Erl and Ricardo Puttini, "Cloud Computing: Concepts Technology & Architecture", PHI.
4. Kailash Jayaswal and Jagannath Kallakurchi, "Cloud Computing Black Book", Willey India.
5. Buyya and Vecchiola, "Mastering Cloud Computing", Morgan Kaufmann *Publishers*.

Course Code	**ITDC0304**
Subject Name	**FULL STACK WEB DEVELOPMENT**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives:

1. Develop proficiency in frontend web development technologies, including HTML, CSS, JavaScript, and React.js, to build responsive and interactive user interfaces.
2. Gain expertise in backend development using Node.js and Express.js, enabling the creation of RESTful APIs, handling HTTP requests, managing middleware, and integrating with databases like MongoDB.
3. Understand and apply the principles of full-stack web development by combining frontend and backend components to build end-to-end applications with features like user authentication, data persistence, and real-time communication.
4. Familiarize with web development best practices, including responsive design, accessibility, performance optimization, version control (Git), and deployment strategies for full-stack applications in production environments.

Course Outcomes:

CO 1. Develop responsive and interactive web applications using HTML, CSS, and JavaScript, adhering to web standards and best practices.

CO 2. Build modern and scalable frontend applications using React.js, including components, state management, routing, and integrating with backend APIs.

CO 3. Design and implement backend services using Node.js and Express.js, with the ability to handle HTTP requests, manage middleware, and integrate with databases like MongoDB.

CO 4. Develop end-to-end full-stack applications by combining frontend and backend technologies, incorporating features like user authentication, data persistence, and RESTful APIs.

Course Outcomes	**Program Outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDC0304														
CO 1	L	H		M				L						
CO 2	L	M						M					L	
CO 3	L		M			H						L		
CO 4	L	M			M									

Course Content:

Unit 1: Why Full Stack Development? Web development vs Full Stack Development, Client-Server architecture, Rules of three-tier architecture, MEAN, MERN, Rails, Django Stack and LAMP, MEAN vs MERN stack, Front End Frameworks and Libraries,
Frontend Fundamentals, Introduction to servers (LAMP/WAMP/XAMP), Difference between global and local servers, Configuration of local servers, Roles of HTTP over the network/Internet, Terminologies (web client, web server, web browser, webpage, website), Client-server architecture, Web hosting, Domain names, URLs

Unit 2: HTML and CSS, HTML basics (structure, elements, attributes), Text formatting (headings, paragraphs, links), Images, tables, lists, HTML forms, HTML5 elements and semantics, CSS basics

(selectors, properties), Text formatting with CSS, Page formatting with CSS, Table formatting with CSS, Border formatting with CSS, List formatting with CSS, CSS layout techniques (floats, positioning, flexbox, grid), CSS3 features (transitions, animations, gradients)

Unit 3: JavaScript, Introduction to JavaScript, Variables, data types, and operators, Control flow (selection and iteration), Functions, Events, Objects, Arrays, Type conversion, Form handling with JavaScript, DOM manipulation, JavaScript ES6 features, Asynchronous JavaScript (callbacks, promises, async/await), AJAX and API calls

Unit 4: Frontend Frameworks, React.js basics (components, state, props, lifecycle), React.js events and routing, Form handling in React, React Hooks, React Redux (state management), Node.js introduction and modules, Express.js basics (routing, middleware), MongoDB basics (CRUD operations), RESTful APIs with Node.js and Express

Unit 5: Web Development Best Practices, Responsive web design, Accessibility, Cross-browser compatibility, Performance optimization, Testing and debugging, Version control (Git), Build tools (Webpack, Babel), Deployment and hosting, Web security

Text/Reference Books:

1. Frank Zammetti, "Modern Full-Stack Development". 2nd Ed. Apress, New York (2022).
2. Shama Hoque, "Full-Stack React Projects". 1st Ed. Packt Publishing, Birmingham (2021).
3. Jon Duckett, "Web Design with HTML, CSS, JavaScript and jQuery". 1st Ed. Wiley, Indianapolis (2018)

Course Code	**ITDC0306**
Subject Name	**BIG DATA ANALYTICS**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives:

1. Discuss the overall process of how data analytics is applied
2. Discuss how data analytics can be used to better address and identify risks
3. Demonstrate the power of data analytics using case studies

Course Outcomes:

CO 1. Understand what Big Data is and why classical data analysis techniques are no longer adequate.
CO 2. Understand the benefits that Big Data can offer to businesses and organizations.
CO 3. Learn conceptually how Big Data is stored.
CO 4. See how Big Data can be analyzed to extract knowledge.

Course Outcomes	**Program Outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDC0306														
CO 1	M	H	L				L						L	
CO 2	H	L	M			M			H					
CO 3	M	L	M				L					H		
CO 4	H	M	H		L				L			H		

Course Content:

Unit-1: Introduction: Big Data Overview, Challenges with Big Data, Traditional Business Intelligence (BI) versus Big Data. Big Data Analytics: The rising and importance of data sciences, Classification of Analytics, Data Sciences, Terminologies in Big Data Environment (In –Memory Analytics, In-Database Processing, Symmetric Multiprocessor system, massively parallel processing, Parallel and distributed system, shared nothing architecture, CAP Theorem), Analytics tools.

Unit-2: Hadoop Overview: Introduction to Hadoop, RDBMS vs Hadoop, key aspects of hadoop, hadoop components, hadoop conceptual layer, high level architecture of hadoop, Hadoop Architecture: Hadoop architecture, Hadoop ecosystem components Hadoop storage: HDFS, Hadoop processing, Map Reduce Framework, Hadoop server roles.

Unit 3: Mathematical concepts in data analytics: Descriptive Statistics, Probability Distributions, Inferential Statistics, Inferential Statistics through hypothesis tests, Regression & ANOVA, Regression, ANOVA (Analysis of Variance). Differentiating algorithmic and model-based frameworks Regression: Ordinary Least Squares, Ridge Regression, Lasso Regression, K Nearest Neighbors Regression & Classification

Unit 4: Supervised Learning with Regression and Classification techniques -1, Bias-Variance

Dichotomy, Model Validation Approaches, Logistic Regression, Linear Discriminant Analysis, Quadratic Discriminant Analysis, Regression and Classification Trees, Support Vector Machines. Supervised Learning with Regression and Classification techniques -2, Ensemble Methods: Random Forest Neural Networks.

Unit 5: Unsupervised Learning and Challenges for Big Data Analytics, Clustering, Associative Rule Mining, Challenges for big data analytics. Prescriptive analytics, creating data for analytics through designed experiments, creating data for analytics through Active learning, creating data for analytics through Reinforcement learning.

Text/Reference Books:

1. S. Acharya and S. Chellappan, "Big data and Analytics", Second Edition, Wiley publication, 2019.
2. Tom White "Hadoop: The Definitive Guide", Forth Edition, O'Reilly Publication, 2015 .
3. David Loshin "Big Data Analytics: From Strategic Planning to Enterprise Integration with Tools, Techniques, NoSQL and Graph" First Edition, Morgan Kaufmann, 2013.
4. Hastie, Trevor, et al. "The elements of statistical learning" Vol. 2. First Edition, Springer NewYork, 2009.
5. Montgomery, Douglas C., and George C. Runger, "Applied statistics and probability for engineers" Sixth Edition, Wiley Publication, 2016

Course Code	**ITDC0308**
Subject Name	**MACHINE LEARNING**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives:

The main objectives of this course include:

1. Providing a comprehensive introduction to various topics in machine learning.
2. Enabling students to design and implement machine learning solutions for classification, regression, and clustering problems.
3. Equipping students with the skills to evaluate and interpret the results of machine learning algorithms.

Course Outcomes:

Students completing the course should be able to:

CO1. Understand basic principles, techniques, and applications of machine learning.

CO2. Gain a broad understanding of machine learning algorithms for data-driven knowledge discovery and program synthesis.

CO3. Recognize the strengths and weaknesses of different machine learning algorithms relative to the characteristics of the application domain.

CO4. Adapt or combine key elements of existing machine learning algorithms to design new algorithms as needed.

Course Outcomes	**Program Outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDC0308														
CO 1	M	H	L				L					H		
CO 2			M			M			M					
CO 3		L					L						L	
CO 4	H	M			L				L					

Course Content:

Unit1: Introduction: Well-Posed Learning Problems, Designing a Learning System, Perspectives and Issues in Machine Learning. Concept Learning and the General-to-Specific Ordering: Introduction, A Concept Learning Task, Concept Learning as Search, FIND-S: Finding a Maximally Specific Hypothesis, Version Spaces and the CANDIDATE-ELIMINATION Algorithm, Basics of machine learning, Classification of machine learning algorithms, Regression algorithms.

Unit-2: Decision Tree Learning: Introduction, Decision Tree Representation, Appropriate problem for Decision tree Learning, The Basic Decision Tree Learning Algorithm, Hypothesis Space Search in Decision Tree Leaning, Inductive Bias in Decision Tree Leaning, Issues in Decision Tree Leaning

Unit 3: Artificial Neural Networks: Introduction, Natural Network Representations, Appropriate Problems for Neural Network Learning, Perceptron, Activation function, Multilayer Network and

the BACKPROPAGATION Algorithm Bayesian Learning: Introduction, Bayes Theorem, Bayes Theorem and Concept Learning, Bayes Optimal Classifier, Native Bayes Classifier, An Example: Learning to Classify Text. Instance- Based Learning: Introduction, K-NEAREST NEIGHBOUR Learning, Distance-Weighted NEAREST NEIGHBOUR Algorithm.

Unit-4: Genetic Algorithms: Motivation, Genetic Algorithms, Hypothesis Space Search, Genetic Programming, Parallelizing Genetic Algorithms. Learning Sets of Rules: Introduction, Sequential Covering Algorithms, Learning Rule Sets: Summary, Learning First-Order Rules, Learning Sets of First-Order Rules: FOIL, Induction as Inverted Deduction, Inverted Resolution.

Unit-5: Support Vector Machine: Maximum margin linear separators, Quadratic Programming Solution to finding maximum margin separators, Kernels for learning non-linear functions, Regression algorithms.

Text/Reference Books:

1. Aurélien Géron, “Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow”, O'Reilly Media, 2019.
2. Andriy Burkov, “Machine Learning Engineering”, True Positive Inc., 2020.
3. Emmanuel Ameisen, “Building Machine Learning Powered Applications”,First Edition, O'Reilly Media, 2020.
4. C. Bishop, “Pattern Recognition and Machine Learning, Springer”,First Edition(Reprint), Springer 2016.
5. Tom M. Mitchell, “Machine Learning”, Mc Graw Hill, 2018.

Course Code	**ITDC0332**
Subject Name	**DISTRIBUTED SYSTEM AND CLOUD COMPUTING LAB**
Contact hours/Credit Scheme (L-T-P-C)	**0-0-2-1**
Pre-requisites	**NONE**

Course Objectives:

1. Understand the basic principle and architecture of cloud computing in different environments.
2. To understand Virtualization Basics, Objectives of Virtualization, and Benefits.
3. An introduction about identity management in cloud and simulate it by using OpenStack.
4. Understand how to create, manage user and group of user's accounts.

Course Outcomes:

CO 1. To understand and apply the knowledge of systems protocols and mechanisms to support Cloud computing.
CO 2. To analyze and implement the concepts of Cloud Computing.
CO 3. To deploy applications on cloud middleware.
CO 4. To understand and resolve security issues in various applications.

Course Outcomes	**Program Outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDC0332														
CO 1	L	H		M									L	
CO 2	L	M	M					M						L
CO 3	L		M			H								
CO 4	L	M	M		M									

List of Practicals:

1. Installation & configuration of Oracle Virtual box for windows XP & android.
2. Service deployment research & uses over cloud-Google app & Amazon web services.
3. Demonstration of Cloud Monitoring tool.
4. Implementation of Resource Provisioning Policy
5. Implementation of resource Scheduling algorithm
6. Implementation of Load balancing algorithm in Cloud computing
7. Implementation of security-based algorithm to protected the cloud platform
8. Performance evaluation of services over cloud-Google App & Amazon web services.
9. Implementation of Dynamic scalability algorithm
10. Managing cloud computing resources.
11. Using existing cloud characteristics and service models.
12. Performance evaluation of services over cloud.
13. Evaluate the case study: Google App Engine, Microsoft Azure

This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents.

Course Code	**ITDC0334**
Subject Name	**FULL STACK WEB DEVELOPMENT LAB**
Contact hours/Credit Scheme (L-T-P-C)	**0-0-2-1**
Pre-requisites	**NONE**

Course Objectives:

After the course completion, the student will be able to perform following practicals

1. Develop proficiency in building responsive and interactive web applications using HTML, CSS, and JavaScript, adhering to web standards and best practices.
2. Gain expertise in developing modern and scalable frontend applications using React.js, including components, state management, routing, and integrating with backend APIs.
3. Learn to design and implement backend services using Node.js and Express.js, handling HTTP requests, managing middleware, and integrating with databases like MongoDB.

Course Outcomes:

CO 1. Students will be able to create a responsive website using HTML, CSS, and JavaScript, incorporating features like navigation menus, image galleries, and form validation.

CO 2. Students will be able to build a full-stack e-commerce application with React.js as the frontend, Node.js and Express.js as the backend, and MongoDB as the database, including features like product listings, shopping carts, and checkout processes.

CO 3. Students will be able to deploy a full-stack application to a cloud platform (e.g., Heroku, AWS, or Google Cloud Platform), ensuring proper configuration and scaling capabilities.

Course Outcomes	**Program Outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDC0334														
CO 1	L	H		M				L						
CO 2	L	M						M					L	
CO 3	L		M			H						L		
CO 4	L	M			M									

List of Practicals:

1. Create a responsive website using HTML, CSS, and JavaScript, incorporating features like navigation menus, image galleries, and form validation.
2. Develop a simple web application using React.js, demonstrating the use of components, state management, and event handling.
3. Build a to-do list application using React.js, implementing CRUD operations and integrating with a backend API.
4. Set up a Node.js server and create RESTful API endpoints for a blog application, handling CRUD operations for blog posts.
5. Implement user authentication and authorization in a Node.js application using JSON Web Tokens (JWT) and bcrypt for password hashing.

6. Connect a Node.js application to a MongoDB database and perform CRUD operations on a collection.
7. Develop a full-stack e-commerce application with React.js as the frontend, Node.js and Express.js as the backend, and MongoDB as the database, including features like product listings, shopping carts, and checkout processes.
8. Build a real-time chat application using Socket.IO, with a React.js frontend and a Node.js backend.
9. Implement server-side rendering (SSR) in a React.js application using Next.js or other frameworks like Gatsby or Remix.
10. Deploy a full-stack application to a cloud platform (e.g., Heroku, AWS, or Google Cloud Platform), ensuring proper configuration and scaling capabilities.ntroduction to CVIP lab or any other tool.

*This is only the suggested list of Practicals. Instructor may frame additional Practicals relevant to the course contents.

Course Code	**ITDC0336**
SUBJECT NAME	**BIG DATA ANALYTICS LAB**
Contact hours/Credit Scheme (L-T-P-C)	**0-0-2-1**
Pre-requisites	**NONE**

Course Objectives:

This course is designed to:

1. Get familiar with Hadoop distributions, configuring Hadoop and performing File management tasks
2. Experiment MapReduce in Hadoop frameworks
3. Implement MapReduce programs in variety applications
4. Explore MapReduce support for debugging
5. Understand different approaches for building Hadoop MapReduce programs for real-time applications

Course Outcomes:

CO 1. Understand what Big Data is and why classical data analysis techniques are no longer adequate.
CO 2. Understand the benefits that Big Data can offer to businesses and organizations.
CO 3. Learn conceptually how Big Data is stored.
CO 4. See how Big Data can be analyzed to extract knowledge.

Course Outcomes	**Program Outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDC0336														
CO 1	M	H	L				L						L	
CO 2	H	L	M			M			H					
CO 3	M	L	M				L					H		
CO 4	H	M	H		L				L			H		

List of Practicals:

1. Install Apache Hadoop
2. Develop a MapReduce program to calculate the frequency of a given word in a given file.
3. Develop a MapReduce program to find the maximum temperature in each year.
4. Develop a MapReduce program to find the grades of student's.
5. Develop a MapReduce program to implement Matrix Multiplication.
6. Develop a MapReduce to find the maximum electrical consumption in each year given electrical consumption for each month in each year.
7. Develop a MapReduce to analyze the weather data set and print whether the day is shinny or cool.
8. Develop a MapReduce program to find the number of products sold in each country by considering sales data containing fields like
   Tranction _Date, Product Price, Payment _Type, Name, City, St State, Country, Account_ Created, Last_Login, Latitude, Longitude
9. Develop a MapReduce program to find the tags associated with each movie by analyzing movie lens data
10. Set up Apache Storm for real-time stream processing, Design and run a simple Storm topology.

This is only the suggested list of Practicals. Instructors may frame additional Practicals relevant to the course contents.

Course Code	**ITDC0338**
Subject Name	**MACHINE LEARNING LAB**
Contact hours/Credit Scheme (L-T-P-C)	**0-0-2-1**
Pre-requisites	**NONE**

Course Objectives:

After completing the lab course, students will be able to:

1. Gain hands-on knowledge of methods and theories in machine learning.
2. Implement decision tree learning, Bayesian learning, and artificial neural networks for real-world problems.
3. Design and implement various classification techniques.

Course Outcomes:

Students completing the lab course should be able to:

CO1. Gain practical experience with machine learning methods and theories.
CO2. Implement decision tree learning, Bayesian learning, and artificial neural networks.
CO3. Design and implement various classification techniques.
CO4. Apply machine learning solutions to real-world problems.

Course Outcomes	**Program Outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDC0338														
CO 1	L	H	L				L					H		
CO 2			M		L	M			M					
CO 3		L					L				L		L	
CO 4	H	M			L				L					

List of Practicals:

1. Study and Implement the Naive Bayes learner using WEKA. (The datasets taken can be: Breast Cancer data file or Reuters data set).
2. Study and Implement the Decision Tree learners using WEKA. (The datasets taken can be: Breast Cancer data file or Reuter‟s data set).
3. Estimate the accuracy of the decision classifier on breast cancer dataset using 5-fold cross-validation. (You need to choose the appropriate options for missing values).
4. Estimate the precision, recall, accuracy, and F-measure of the decision tree classifier on the text classification task for each of the 10 categories using 10-fold cross-validation.
5. Develop a machine learning method to classifying your incoming mail.
6. Develop a machine learning method to Predict stock prices based on past price variation.
7. Develop a machine learning method to predict how people would rate movies, books, etc.
8. Develop a machine learning method to Cluster gene expression data, how to modify existing methods to solve the problem better
9. Select two datasets. Each dataset should contain examples from multiple classes. For training purposes assume that the class label of each example is unknown (if it is known, ignore it). Implement the K-means algorithm and apply it to the data you selected. Evaluate performance by measuring the sum of Euclidean distance of each example from its class center. Test the performance of the algorithm as a function of the parameter k.

10. Implement the EM algorithm assuming a Gaussian mixture. Apply the algorithm to your datasets and report the parameters you obtain. Evaluate performance by measuring the sum of Mahalanobis distance of each example from its class center. Test performance as a function of the number of clusters.
11. Suggest and test a method for automatically determining the number of clusters. Using a dataset with known class labels compare the labeling error of the K-means and EM algorithms. Measure the error by assigning a class label to each example. Assume that the number of clusters is known.

This is only the suggested list of Practicals. Instructors may frame additional Practicals relevant to the course contents.

Departmental Electives

DEPARTMENTAL ELECTIVES I

S.No.	Subject Code	Subject Name	L	T	P	Credits
1	ITDE0351	Data Mining	3	0	0	3
2	ITDE0353	Computer Graphics and Applications	3	0	0	3

Course Code	**ITDE0351**
Subject Name	**DATA MINING**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives:

The student should be made to:

1. Provide an overview of data mining.
2. Understanding of mathematical concepts and algorithms used in data mining.
3. To introduce the foundational concepts and practical implications of Data Mining Techniques
4. To survey the state-of-the-art advancements in theories and applications of Data Mining
5. Provide programming experience for developing and implementing data mining applications.

Course Outcomes:

CO 1. To effectively carry out further research on Data Mining techniques
CO 2. To effectively develop new applications based on Data Mining Concept
CO 3. To introduce students to the basic concepts and techniques of Data Mining.
CO 4. To develop skills of using recent data mining software for solving practical problems.
CO 5. To gain experience of doing independent study and research.

Course Outcomes	**Program Outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDE0351														
CO 1	M				L							L		L
CO 2		H						M						
CO 3			M			H							L	
CO 4		H			H				L					

Course Content:

Unit 1: Data Mining: Introduction, KDD vs Data Mining, DBMS vs DM, Other Related Areas, DM Techniques, Other Mining Techniques, Issues and Challenges in DM, DM Applications- Case Studies, Creating data for data mining. Data mining model creation. Selecting data mining algorithm. Understanding data mining tools. Mapping Mining Structure to Source Data columns. Using Cube Sources. Configuring Algorithm parameters.

Unit 2: Data Classification: decision trees, choosing the splitting attribute, information gain and gain ratio, handling numeric attributes (finding best split), dealing with missing values, pruning (pre-pruning, post-pruning, estimating error rates), from trees to rules, Classification: naive Bayes classifier, neural networks, support vector machines

Unit 3: Clustering Techniques: Basic Concepts, Types of Clustering Methods, Hierarchical Clustering, Categorical Clustering, K-means Algorithm, Agglomerative Hierarchical Clustering, Proximity between clusters, DBSCAN, Partitioning Algorithms, k-Medoid Algorithms, CLARA, CLARANS, DBSCAN, BIRCH, CURE, STIRR, ROCK, CACTUS

Unit 4: Advanced Concepts: Mining Object: Spatial, Multimedia, Text and Web data, Web Mining, Web Content Mining, Web Structure Mining, Web Usage Mining, Text Mining, Unstructured Text, Episode Rule Discovery for Texts, Hierarchy of Categories, Time series data mining, Mining sequence patterns in Transactional databases.

Text/Reference Books:

1. Ian H. Witten and Eibe Frank, Data Mining: Practical Machine Learning Tools and Techniques (Fourth Edition), Morgan Kaufmann, 2016.
2. M.H.Dunham, "Data Mining: Introductory and Advanced Topics", 1st Ed., Pearson Education, 2013.
3. Jiawei Han, Micheline Kamber, "Data Mining Concepts & Techniques", 4th Ed., Elsevier, 2023.
4. Sam Anahory, Dennis Murray, "Data Warehousing in the Real World : A Practical Guide for Building Decision Support Systems", Pearson, 1st Edition, 2015.
5. Mallach, "Data Warehousing System", McGraw –Hill, 1st Edition, 2000.
6. Alex Berson, Stephen J. Smith, "Data Warehousing, Data Mining, & Olap", Tata McGraw-Hill Education, 1st Edition 2013.

Course Code	**ITDE0353**
Subject Name	**COMPUTER GRAPHICS AND APPLICATIONS**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives:

1. Study the works of contemporary artists, designers as well as the masters in the field and discuss and enrich their vocabulary of design.
2. Learn ways to apply aesthetic sensibilities into their works and explore ways to balance between formal theories with practical applications.
3. Gain skill to use the digital tools as a powerful means of communication for creation, modification & presentation.

Course Outcomes:

CO1. Undertsand the underlying architecture, algorithms, mathematical concepts, supporting computer animation.
CO2. Enhance their perspective of modern computer system with modeling, analysis and interpretation of 2D and 3D
CO3. Apply the graphic design theory and principles as they relate to current trends in visual communication
CO4. Analyze the information to make compelling and experimental visual expressions for presentation

Course Outcomes	**Program Outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDE0353														
CO 1		M			L							H		
CO 2	L		M					M						H
CO 3					M				L					
CO 4		L											H	

Course Content:

Unit 1: Introduction: Introduction and Application areas of Computer Graphics, overview of graphics systems, video display devices, raster-scan systems, random scan systems, graphics monitors and work stations and input devices**.**

Output primitives: Points and lines, line drawing algorithms, mid-point circle and ellipse algorithms.

Filled area primitives: Scan line polygon fill algorithm, boundary-fill and flood-fill algorithms

Unit 2: 2-D geometrical transforms: Translation, scaling, rotation, reflection and shear transformations, matrix representations and homogeneous coordinates, composite transforms, transformations between coordinate systems.

2-D viewing: The viewing pipeline, viewing coordinate reference frame, window to view-port coordinate transformation, viewing functions, Cohen-Sutherland and Cyrus-beck line clipping algorithms, Sutherland – Hodgeman polygon clipping algorithm**.**

Unit 3: 3-D object representation: Polygon surfaces, quadric surfaces, spline representation, Hermite curve, Bezier curve and B-Spline curves, Bezier and B-Spline surfaces. Basic illumination models, polygon rendering methods.

3-D Geometric transformations: Translation, rotation, scaling, reflection and shear transformations, composite transformations.

3-D viewing: Viewing pipeline, viewing coordinates, view volume and general projection transforms and clipping.

Unit 4: Visible surface detection methods: Classification, back-face detection, depth-buffer, scan-line, depth sorting, BSP-tree methods, area sub-division and octree methods.

Unit 5: Computer animation: Design of animation sequence, general computer animation functions, raster animation, computer animation languages, key frame systems, motion specifications.

Text/Reference Books:

1. Donald Hearn, M. Pauline Baker, Warren Carithers (2013). Computer Graphics with Open GL, 4th Edition. Pearson Education India.
2. Evans, Poppy, SherinAaris. (2013). The Graphic Design: Reference & Specification Book Sixth edition 1998. US: Rockport Publishers.
3. Ellen, Phillips & Jennifer, C. P. (2015). Graphic Design: The New Basics: 2nd Edition. UK: Princeton Architectural Press
4. David, Dabner, Sanra, Stewart & Eric, Zempol. (2014). Graphic Design Shool. Thames & Hudson

DEPARTMENTAL ELECTIVES II

S.No.	Subject Code	Subject Name	L	T	P	Credits
1	ITDE0352	Mobile Communication	3	0	0	3
2	ITDE0354	Principles of Programming Languages	3	0	0	3
3	ITDE0356	Business Intelligence	3	0	0	3

Course Code	**ITDE0352**
Subject Name	**MOBILE COMMUNICATION**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives:

1. To learn about the concepts and principles of mobile computing;
2. To explore both theoretical and practical issues of mobile computing;
3. To develop skills of finding solutions and building software for mobile computing applications.

Course Outcomes:

CO 1. To understand the infrastructure to develop mobile communication systems (cellular theory) and the characteristics of different multiple access techniques in mobile communication.

CO 2. To visualize the various important steps in GSM communication.

CO 3. To examine the important aspects of Mobile Adhoc Networks.

CO 4. To analyze and compare the various wireless communication technologies.

Course Outcomes	Program Outcomes													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDE0352														
CO 1		M	M											L
CO 2		M		H				H						
CO 3			H	M	M									
CO 4	H	H			M	H	H		H		H			

Course Content:

Unit 1: Introduction, Applications, History of wireless communication, A market of mobile communication, Reference model

Unit 2: Wireless transmission: Frequencies for radio transmission, Signals, Antennas, Signal propagation, Multiplexing, Modulation, Spread spectrum, Cellular System

Unit 3: Medium access control (MAC): Motivation of specialized MAC, Space Division Multiple Access (SDMA), Frequency division multiple access (FDMA), Time Division Multiple Access (TDMA), Carrier Sense Multiple Access (CDMA). Telecommunication system: Global System for Mobile Communication (GSM), Digital Enhanced Cordless Telecommunications (DECT), Terrestrial Trunked Radio (TETRA), UMTS and International Mobile Telecommunications (IMT)-2000

Unit 4: Satellite and Broadcast System: Applications of Satellite and broadcasting system, GEO, LEO, MEO, Routing, Localization, Handover, Cyclic repetition of data, Digital audio broadcasting, Digital video broadcasting, Convergence of broadcasting and mobile communication. Wireless LAN: Infrared vs radio transmission, Infrastructure and ad-hoc network, IEEE 802.11, HIPERLAN, Bluetooth

Unit 5: Mobile network layer: Mobile IP, Dynamic host configuration protocol, Mobile Ad-hoc network, Mobie transport layer: Traditional TCP, Classical TCP improvements, TCP over 2.5/3G wireless network, Performance enhancing proxies, Support of mobility: File system, World wide web, Wireless application protocols, i-mode, SyncML, WAP 2.0. The architecture of future network.

Text/Reference Books:

1. Jochen Schiller, "Mobile Communications", Pearson Education, Second Edition 2002.
2. C.K. Toh, "Ad Hoc Mobile Wireless Networks: Protocols and Systems", Pearson Education, 2002.
3. William Stallings, "Wireless Communications and Networks", Pearson Education 2002.

Course Code	**ITDE0354**
Subject Name	**PRINCIPLES OF PROGRAMMING LANGUAGES**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives:

1. To introduce the major programming paradigms, and the principles and techniques involved in design and implementation of modern programming languages.
2. To introduce notations to describe syntax and semantics of programming languages.
3. To analyze and explain behavior of simple programs in imperative languages using concepts such as binding, scope, control structures, subprograms and parameter passing mechanisms.
4. To introduce the concepts of ADT and object oriented programming for large scale software development.

Course Outcomes:

CO 1. Understand the fundamental concepts of most programming languages & the tradeoff between language design and implementation.

CO 2. Assess programming languages critically and scientifically.

CO 3. Understand different programming paradigms: analyze the principles of imperative, object-oriented, functional and logic programming.

CO 4. Design a new programming language in principle.

Course Outcomes	**Program Outcomes**

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDE0354														
CO 1	H	M		L	M	L								L
CO 2		H		H	M	H	M						M	M
CO 3			M		M		M	H					M	
CO 4			H	M					H	H	H	M		H

Course Content:

Unit 1: Introduction: Characteristics of programming Languages, Factors influencing the evolution of programming language, developments in programming methodologies, desirable features and design issues. Programming language processors: Structure and operations of translators, software simulated computer, syntax, semantics, structure, virtual computers, binding and binding time.

Unit 2: Elementary and Structured Data Types: Data object variables, constants, data types, elementary data types, declaration, assignment and initialization, enumeration, characters, strings. Structured data type and objects: Specification of data structured types, vectors and arrays, records, variable size data structure, pointers and programmer constructed data structure, Sets files. Sub Program and programmer defined data types: Evolution of data types, abstractions, encapsulations, information hiding, sub programmes, abstract data types.

Unit 3: Sequence Control: Implicit and Explicit sequence control, sequence control with within expression and statements, recursive sub programmes, exception handling, co routines, Scheduled sub programmes, concurrent execution. Data control referencing environments, static and dynamic scope, local data local data referencing environment, shared data: Explicit common environment dynamic scope parameter passing mechanism.

Unit 4: Storage Management: Major run time requirements, storage management phases, static storage management, stack based, heap based storage management. Syntax and translation: General syntactic criteria, syntactic element of a language, stages in translation, formal syntax and semantics.

Unit 5: Operating and Programming Environment: Batch Processing Environments, Embedded system requirements, Theoretical models, Introduction to Functional Programming, Lambda calculus, Data flow language and Object Oriented language, Comparison in various general and special purpose programming languages e.g. Fortran, C, Pascal, Lisp, etc.

Text/Reference Books:

1. Rajan, Hridesh. An Experiential Introduction to Principles of Programming Languages. United States: MIT Press, 2022.
2. Dowek, Gilles. Principles of Programming Languages. Germany: Springer London, 2009.
3. Terrance W Pratt, "Programming Languages: Design and Implementation" PHI, 2000.
4. Sebesta, "Concept of Programming Language", Addison Wesley, 2003.
5. E Horowitz, "Programming Languages", 2nd Edition, Addison Wesley, 2005.
6. Kenneth C Louden, Programming Languages: Principles and Practice, 3rd Edn., Cengage. Learning, 2011

Course Code	**ITDE0356**
Subject Name	**BUSINESS INTELLIGENCE**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives: The student should be made to:

1. Be exposed with the basic rudiments of business intelligence system
2. Understand the modeling aspects behind Business Intelligence
3. Understand of the business intelligence life cycle and the techniques used in it
4. Be exposed with different data analysis tools and techniques

Course Outcomes:

CO 1. Understand Fundamental concepts of BI and Analytics and application of BI Key Performance indicators.

CO 2. Design of Dashboards, Implementation of Web Analytics.

CO 3. Understand Utilization of Advanced BI Tools and their Implementation.

CO 4. Implementation of BI Techniques and BI Ethics.

Course	Program Outcomes

Outcomes														
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDE0356														
CO 1	H	M	M	L										
CO 2	L			H	H								L	
CO 3						M	M	H				H		M
CO 4					L		M	H		H	L	H	L	

Course Content:

Unit 1: Business Intelligence: Introduction, Effective and timely decisions – Data, information and knowledge – Role of mathematical models – Business intelligence architectures: Cycle of a business intelligence analysis – Enabling factors in business intelligence projects – Development of a business intelligence system – Ethics and business intelligence.

Unit 2: Data Mining: Meaning and purpose. Creating data for data mining. Data mining model creation. Selecting data mining algorithm. Understanding data mining tools. Mapping Mining Structure to Source Data columns. Using Cube Sources. Configuring Algorithm parameters.

Unit 3: Understanding OLAP: Understanding Multidimensional Analysis Concepts: Attributes, Hierarchies and Dimensions in data Analysis. Understanding Dimensional Data Warehouse: Fact Table, Dimension Tables, Surrogate Keys and alternative Table Structure.

Unit 4: Advanced Measures and Calculations**,** Advanced Dimensional Design**,** Retrieving Data from Analysis Services

Text/Reference Books:

1. Carlo Vercellis (2011). "Business Intelligence: Data Mining and Optimization for Decision Making". John Wiley & Sons.
2. David Loshin (2012). "Business Intelligence: The Savvy Manager's Guide". Newnes.
3. Elizabeth Vitt, Michael Luckevich, Stacia Misner (2010). "Business Intelligence". O'Reilly Media, Inc.
4. Rajiv Sabhrwal, Irma Becerra-Fernandez (2010). "Business Intelligence". John Wiley & Sons.
5. Swain Scheps (2013). "Business Intelligence for Dummies". Wiley.

DEPARTMENTAL ELECTIVES III

S.No.	Subject Code	Subject Name	L	T	P	Credits
1	ITDE0451	Cyber Security	3	0	0	3
2	ITDE0453	Soft Computing	3	0	0	3
3	ITDE0455	Internet of Things (IoT)	3	0	0	3

Course Code	**ITDE0451**
Subject Name	**CYBER SECURITY**

Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives:

1. To create awareness of Information security issues and challenges in the IT environment.
2. To impart basic knowledge and skills to protect one's personal IT assets.
3. To learn the techniques needed for providing protection and security to our personal data and information resources.
4. To experiment and learn the skills to provide protection and security to organizational data and information to build a secured IT infrastructure in the companies.

Course Outcomes:

CO 1. Understand the fundamental concepts in cyber security and distinguish among the attacks, threats and vulnerabilities.
CO 2. Apply cyber security solutions and information assurance.
CO 3. Classify and distinguish different laws and defense methods
CO 4. Distinguish different crimes and frauds.

Course Outcomes	**Program Outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDE0451														
CO 1	H		H	M									L	M
CO 2	M	M	H	M				H					M	L
CO 3	L		M										H	M
CO 4					L		M	H		H	L	H	L	

Course Content:

Unit 1: Cyber World: An Overview: The internet and online resources, Security of information, Digital signature. An Overview of Cyber Law: Introduction about cyberspace, Regulation of cyberspace – introducing cyber law, UNCITRAL Model Law on Electronics Commerce 1996.

An Overview of Cyber Crimes: Defining Crime, Crime in context of the Internet –Actus Rea/Mens Rea, Types of crime in Internet, Computing damage in Internet crime. An Overview of Indian Penal Law & Cyber Crimes: Fraud, Hacking, Mischief, Trespass, Defamation, Stalking, Spam.

Unit 2: An Overview of Human Rights Issues in Internet: Freedom of Expression in Internet, Issues of Censorship, Hate speech, Sedition, Libel, Subversion, Privacy Issues, International Positions on Free Speech in Internet

Unit 3: An Overview of Electronic Contracts: The Indian Law of Contract, Construction of Electronic Contracts, Issues of Security Issues of Privacy Technical Issues in Cyber Contracts

Types of Electronic Contracts: Employment Contracts Consultant Agreements Contractor Agreements Sales, Reseller and Distributor Agreements Non-Disclosure Agreements, Software Development & Licensing Agreements Shrink Wrap Contract, Source Code Escrow Agreements.

Unit 4: An Overview of CopyRight in Information Technology: Understanding the technology of Software, Software - Copyrights vs. Patents debate, Authorship and Assignment Issues, Commissioned Work and Work for Hire, Idea/Expression dichotomy, Copyright in Internet.

Text/Reference Books:

1. Vivek Sood, Cyber law Simplified, Tata Mcgraw-Hill Publishing, 202.
2. Chris Reed and john Angel, Cyber law, Oxford University Press, 2023.
3. Sudhir Naib, The Information Technology Act, 2005: A Handbook, OUP, New York, 2022
4. S. R. Bhansali, Information Technology Act, 2000, University Book House Pvt. Ltd., 2023

Course Code	**ITDE0453**
Subject Name	**SOFT COMPUTING**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives:

1. Motivation and historical background of Soft Computing.
2. Application of Fuzzy logic.

3. Biologically inspired algorithm such as neural networks, genetic algorithms, ant colony optimization, and bee colony optimization.
4. Hybrid systems of neural network, genetic algorithms and fuzzy systems.

Course Outcomes:

CO 1. Understand and describe soft computing techniques and their roles in building intelligent machines

CO 2. Recognize the feasibility of applying a soft computing methodology for a particular problem

CO 3. Effectively use existing software tools to solve real problems using a soft computing approach

CO 4. Evaluate and compare solutions by various soft computing approaches for a given problem.

Course Outcomes	Program Outcomes													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDE0453														
CO 1		M			M									M
CO 2			M				L						L	
CO 3			L	L										M
CO 4					M									

Course Content:

Unit 1: Artificial Neural Networks: Basic concepts - Single layer perception - Multilayer Perception - Supervised and Unsupervised learning – Back propagation networks - Kohnen'sself organizing networks - Hopfield network.

Unit 2: Fuzzy Systems: Fuzzy sets and Fuzzy reasoning - Fuzzy matrices - Fuzzy functions - Decomposition -Fuzzy automata and languages - Fuzzy control methods - Fuzzy decision making.

Unit 3: Neuro - Fuzzy Modeling: Adaptive networks based Fuzzy interface systems - Classification and Regression Trees -Data clustering algorithms - Rule based structure identification - Neuro-Fuzzy controls -Simulated annealing – Evolutionary computation.

Unit 4: Genetic Algorithms: Survival of the Fittest - Fitness Computations - Cross over - Mutation - Reproduction -Rank method - Rank space method.

Unit 5: Soft computing And Conventional AI: AI search algorithm - Predicate calculus - Rules of interference – Semantic networks -Frames - Objects - Hybrid models - Applications.

Text/Reference Books:

1. Ranjit Panigrahi, Samarjeet Borah, Applied Soft Computing: Techniques and Applications. United Kingdom: Apple Academic Press, 2022.
2. Saroj Kaushik and Sunita Tewari, Soft Computing, McGraw Hill Education, 2018.
3. Sameer Roy and Udit Chakraborty, Soft Computing, Pearson, 2013.

4. Jang J.S.R., Sun C.T. and Mizutani E, "Neuro-Fuzzy and Soft computing", Prentice Hall 1998.
5. LaureneFausett, "Fundamentals of Neural Networks", Prentice Hall, 1994.
6. George J. Klir and Bo Yuan, "Fuzzy sets and Fuzzy Logic", Prentice Hall, USA 1995.
7. N. J. Nelsson, "Artificial Intelligence - A New Synthesis", Harcourt Asia Ltd., 1998.
8. D.E. Goldberg, "Genetic Algorithms: Search, Optimization and Machine Learning", Addison Wesley, N.Y, 1989.

Course Code	**ITDE0455**

Subject Name	INTERNET OF THINGS (IoT) CONCEPTS
Contact hours/Credit Scheme (L-T-P-C)	3-0-0-3
Pre-requisites	NONE

Course Objectives:
Students will be explored to the interconnection and integration of the physical world and the cyber space. They are also able to design & develop IOT Devices.

Course Outcomes:

CO 1. Understand the application areas of IOT
CO 2. Able to realize the revolution of Internet in Mobile Devices, Cloud & Sensor Networks
CO 3. Understand building blocks of Internet of Things and characteristics.

Course Outcomes	Program Outcomes													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDE0455														
CO 1	H	M		M		L		L		H				
CO 2		M	L	M							L	M		M
CO 3		M		H	L				M	M	L	H		L

Course Content:

Unit 1: Introduction & Concepts: Introduction to Internet of Things, Physical Design of IOT, Logical Design of IOT, IOT Enabling Technologies, IOT Levels.

Unit 2: Domain Specific IOTs: Home Automation, Cities, Environment, Energy, Retail, Logistics, Agriculture, Industry, Health & Life Style.

Unit 3: M2M & System Management with NETCONF-YANG: M2M, Difference between IOT and M2M, SDN and NFV for IOT, Software defined Networking, Network Function Virtualization, Need for IOT Systems Management, Simple Network Management Protocol, Limitations of SNMP, Network Operator Requirements, NETCONF, YANG, IOT Systems management with NETCONF-YANG.

Unit 4: Developing Internet of Things & Logical Design using Python: Introduction, IOT Design Methodology, Installing Python, Python Data Types & Data Structures, Control Flow, Functions, Modules, Packages, File Handling, Date/ Time Operations, Classes, Python Packages

Unit 5: IoT Physical Devices & Endpoints: What is an IOT Device, Exemplary Device, Board, Linux on Raspberry Pi, Interfaces, and Programming & IOT Devices.

Text/Reference Books:

1. Vijay Madisetti, Arshdeep Bahga, “Internet of Things A Hands-On- Approach”,2014, ISBN: 978 0996025515
2. Adrian McEwen, “Designing the Internet of Things”, Wiley Publishers, 2013, ISBN: 978-1-118-43062-0

3. Daniel Kellmereit, “The Silent Intelligence: The Internet of Things”. 2013, ISBN 0989973700

DEPARTMENTAL ELECTIVES IV

S.No.	Subject Code	Subject Name	L	T	P	Credits
1	ITDE0461	Wireless Communication for Beyond 5G Networks and IoT	3	0	0	3
2	ITDE0463	Image Processing	3	0	0	3
3	ITDE0465	Prompt Engineering	3	0	0	3

Course Code	**ITDE0461**
Subject Name	**WIRELESS COMMUNICATION FOR BEYOND 5G NETWORKS AND IOT**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives:

1. Able to understand different optical sources and wireless communications.
2. Recognize the feature and importance of different wireless communication and different modulation schemes.
3. Able to analyse the performance of different Optical wireless communication.

Course Outcomes:

CO 1. Develop the understanding the Optical sources and Optical wireless communication systems
CO 2. Describe different modulation schemes.
CO 3. Describe the concept of different types of detectors, Indoor and Outdoor Optical wireless communication channel Modelling
CO 4. Analyse system performance for Indoor and Outdoor Optical wireless communication.

Course Outcomes	Program outcomes													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDE0461														
CO 1	M				M				L		L			M
CO 2	L	L			M						L			
CO 3	L	L			L		L		L					L
CO 4	M	H	H		H	M					M			M

Course Content:

Unit 1: Introduction: Optical Wireless Communication Systems; Existing wireless Access Schemes, OWC/Radio Comparison, Potential OWC Application Areas

Optical Sources: LEDs and Lasers (Internal and External Quantum Efficiency, Power and Luminous Efficiency, and Modulation Bandwidth)

Detectors : PIN and APD Photodetector, Photodetection Techniques, Photodetection Noise

Unit 2: Indoor Optical Wireless Communication Channel Modelling: LOS Propagation Model, Non-LOS Propagation Model, Interference from other other Light sources

Outdoor Optical Wireless Communication Channel Modelling: Atmospheric Channel Loss, Beam Divergence, Pointing Loss, Different Atmospheric Turbulence Models

Unit 3: Underwater Optical Wireless Communication Channel Modelling: Absorption, scattering, Turbulence, Multipath interference, Physical obstruction, and Background noise.

Unit 4: Modulation Schemes Digital Baseband Modulation Techniques like PAM, PPM, PIM etc., Multi-carrier Modulation (OFDM) for OWC, Color Shift Keying, NOMA etc.

Unit 5: System Performance Analysis: Indoor OWC links Effect of Ambient Light Sources on Indoor OWC Link Performance, Link Performance for Multipath Propagation

System Performance Analysis: Outdoor OWC links FSO Link Performance under the Effect of Atmospheric turbulence, Atmospheric Turbulence-Induced Penalty and mitigation strategies

Text/Reference Books:

1. Shlomi Arnon, John Barry, George Karagiannidis, Robert Schober, Murat Uysal, "Advanced Optical Wireless Communication Systems", 1st Edition, Cambridge University Press, 2012.
2. Z. Ghassemlooy, W. Popoola, S. Rajbhandari, "Optical Wireless Communications: System and Channel Modelling with MATLAB", 2nd Edition, CRC Press, 2019 .
3. Gerd Keiser, "Optical Communications", 5th Edition, McGraw-Hill Education, 2013.

Course Code	**ITDE0463**
Subject Name	**IMAGE PROCESSING**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives:

1. Able to learn the basic theory and algorithms that are widely used in digital image processing.
2. Involves processing and transforming large, incomprehensible datasets into a format more easily understood by humans
3. Learn current technologies and issues that are specific to image processing systems. They should develop hands-on experience to process images.

Course Outcomes:

After the completion of the course student will be able to:

CO 1. Explain the fundamentals of digital image and its processing
CO 2. Perform image enhancement techniques in spatial and frequency domain.
CO 3. Elucidate the mathematical modeling of image restoration and compression
CO 4. Apply the concept of image segmentation.
CO 5. Describe pattern recognition techniques.

Course Outcomes	**Program Outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDE0463														
CO 1		M			L								L	
CO 2	L		M					M						
CO 3					M				L					
CO 4		L												M

Course Content:

Unit 1: Introduction and fundamental to Digital Image Processing: Origin of Digital Image Processing, Fundamental steps in Digital Image Processing, Components of Digital Image Processing System, Image sensing and acquisition, Image sampling, quantization and representation, Basic relationship between pixels, Color Fundamentals, Color models, Basis of full color image processing, Color Transformations

Unit 2: Image Enhancement in spatial and frequency domain: Basic grey level transformation, Histogram processing, Basics of Spatial filtering, Smoothing and Sharpening spatial filters,

Introduction to Fourier Transform and the Frequency Domain, Discrete Fourier Transform, Smoothing and Sharpening Frequency Domain filters.

Unit 3: Image Restoration and Image compression: IImage Degradation/Restoration Process, Noise models, restoration in presence of noise, Inverse filtering, Minimum Mean Square Filtering, Geometric menu filter, Geometric transformations, Fundamentals, Image compression models, Error free compression, Lossy compression.

Unit 4: Image segmentation, Representation, Description and Recognition: Detection of Discontinuities, Edge linking and boundary detection, Threshold, Region oriented segmentation, Representation-chain codes, polygonal approximation and skeletons, Boundary descriptors, simple descriptors, shape numbers, Regional descriptors, topological descriptors.

Unit 5: Pattern Recognition: Pattern and pattern classes, Tree classifiers: Decision trees, random forests, Parametric techniques: Maximum likelihood Estimation, Non-Parametric techniques: Kernel Density estimators, Nearest Neighbor methods

Text/Reference Books:

1. Rafael C. Gonzalez, Richard E. Woods - "Digital Image Processing", 4th Edition, Pearson Education, 2018
2. Ian Pointer - "Practical Computer Vision Applications Using Deep Learning with CNNs: With Detailed Examples in Python Using TensorFlow and Kivy", 1st Edition, Apress, 2019.
3. A.K. Jain - "Digital Image Processing", 1st Edition, PHI, 1989.
4. Madhuri A. Joshi - "Digital Image Processing: An Algorithmic Approach", 1st Edition, PHI Learning, 2006.

Course Code	**ITDE0465**
Subject Name	**PROMPT ENGINEERING**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives:

1. Be exposed with the basic rudiments of business intelligence system
2. understand the modeling aspects behind Business Intelligence
3. understand of the business intelligence life cycle and the techniques used in it
4. Be exposed with different data analysis tools and techniques

Course Outcomes:

CO 1. Understanding large language models and their limitations
CO 2. Prompt formulation and refinement
CO 3. Developing and using prompt templates
CO 4. Prompt and response evaluation
CO 5. Responsible innovation and broader societal implications of LLMs.

Course Outcomes	**Program Outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDE0465														
CO 1	H	M	M	L										
CO 2	L			H	H				L					M
CO 3						M		M	M			M		
CO 4					L				L	H				
CO 5							H	M	H		M	M		H

Course Content:

Unit 1: Intro To Prompt Engineering with ChatGPT: Intro to natural language processing (NLP) and prompt engineering, Prompt engineering application in ChatGPT, Understanding the benefits and challenges of prompt engineering, Real-world examples of prompt engineering in action.

Understanding The Limits of Prompt Engineering: Recognizing when using prompt solutions is not appropriate, Identifying the limitations of prompt solutions, Activity: Analyzing the limitations of prompt solutions.

Unit 2: Writing Powerful Prompts: Intro to task-specific prompts, Techniques for creating powerful, task-specific prompts, Best practices for writing effective prompts, Exercise in writing prompts.

Unit 3: Quality Control & Testing of Existing Prompts: Understanding the importance of evaluating and testing existing prompt solutions, Techniques for evaluating and testing prompt solutions, Best practices for refining existing prompt solutions or developing new Ones, Activity: Evaluating and refining existing prompt solutions.

Unit 4: Automating Prompt Engineering Solutions: Overview of automated techniques used in prompt engineering for ChatGPT, Benefits and limitations of automated prompt engineering for ChatGPT, Best practices for using automated techniques in developing efficient, effective, and quality-controlled prompt solutions, Activity: Using automated techniques to develop prompt solutions.

Unit 5: Ethical Considerations in Prompt Engineering; Understanding the ethical considerations and potential biases in prompt engineering, Best practices for ensuring fairness and accountability in prompt engineering, Discussion in analyzing ethical considerations in prompt engineering for ChatGPT

Text/Reference Books:

1. Carlo Vercellis, "Business Intelligence: Data Mining and Optimization for Decision Making", 1st Edition, John Wiley & Sons, 2009.
2. David Loshin, "Business Intelligence: The Savvy Manager's Guide", 2nd Edition, Morgan Kaufmann, 2013.
3. Elizabeth Vitt, Michael Luckevich, Stacia Misner, "Business Intelligence", 1st Edition, O'Reilly Media, Inc., 2010.
4. Rajiv Sabherwal, Irma Becerra-Fernandez, "Business Intelligence: Practices, Technologies, and Management", 1st Edition, John Wiley & Sons, 2010.
5. Swain Scheps, "Business Intelligence for Dummies", 2nd Edition, Wiley, 2011..

DEPARTMENTAL ELECTIVES V

S.No.	Subject Code	Subject Name	L	T	P	Credits
1	ITDE0452	DevOps/MLOps	3	0	0	3
2	ITDE0454	Mobile Application Development	3	0	0	3
3	ITDE0456	High Performance Computing	3	0	0	3

Course Code	**ITDE0452**
Subject Name	**DEVOPS/MLOPS**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	1. Knowledge of Linux Operating system, installation and configuration of services and command line basics. 2. Basics of Computer Networks and Software 3. Software Development Life cycle.

Course Objectives:

1. The objective of this course is to understand the fundamentals of DevOps engineering.
2. Be fully proficient with DevOps terminologies, concepts, benefits, and deployment options to meet real world software development requirements.

Course Outcomes:

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

CO 1. Understand the procedures to facilitate more dependable software releases.
CO 2. Apply DevOps principles to meet software development requirements.
CO 3. Become proficient in implementing different practices aiming to increase the efficiency of the software development life cycle.

Course Outcomes	Program Outcomes													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDE0452														
CO 1	H	M	M	L		M	L		M					L
CO 2					M			L	L	M			H	
CO 3		L			H	H		L			H	H		

Course Content:

Unit 1: Introduction to DevOps: Phases of Software Lifecycle, Minimum Viable Product (MVP) & Cross-functional Teams, Lean, ITIL, Agile development methodologies, DevOps as a prominent culture to achieve agility in the software development process, History of DevOps, DevOps Stakeholders, Goals, Important terminology, DevOps and Agile, DevOps Tools, Configuration management, Continuous Integration and Deployment (CI/CD), DevOps application delivery.

Unit 2: DevOps Principles and Practices: 7 C's of DevOps Lifecycle for Business Agility, DevOps and Continuous Testing, How to Choose Right DevOps Tools, Challenges with DevOps Implementation, Must Do Things for DevOps, Mapping My App to DevOps - Assessment, Definition, Implementation, Measure and Feedback. Version Control, Continuous Integration, Creating Views and Jobs in Jenkins,
Continuous Deployment and Testing.

Unit 3: Continuous Management: Overview of Infrastructure as a code, Benefits of Infrastructure as Code, The Four Key Metrics, Three Core Practices for Infrastructure as Code, The Parts of an Infrastructure System, Infrastructure Platforms, Infrastructure Resources, Compute Resources, Storage Resources, Network Resources

Unit 4: Puppet Architecture, The Puppet Server, setting up the Puppet Agent, Performance Optimizations, Completing the stack with PuppetDB, The PuppetCA
Ansible: Ansible Architecture, Ansible and Infrastructure Management, Local Infrastructure Development: Ansible and Vagrant.

Text/Reference Books:

1. Jennifer Davis, Ryn Daniels, "Effective DevOps", 1st Edition, O'Reilly Media, 2016.
2. Gene Kim, Jez Humble, Patrick Debois, John Willis, "The DevOps Handbook", 2nd Edition, IT Revolution Press, 2021.
3. Betsy Beyer, Chris Jones, Jennifer Petoff, Niall Richard Murphy, "Site Reliability Engineering: How Google Runs Production Systems", 1st Edition, O'Reilly Media, 2016.

Course Code	**ITDE0454**
Subject Name	**MOBILE APPLICATION DEVELOPMENT**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives:

1. Introduction and characteristics of mobile applications.
2. Application models of mobile application frameworks.
3. Managing application data and User-interface design for mobile applications.
4. Integrating networking, the OS and hardware into mobile-applications.
5. Addressing enterprise requirements in mobile applications – performance, scalability, modifiability, availability and security.

Course Outcomes:

CO 1. Understand technology and business trends impacting mobile applications

CO 2. Be competent with the characterization and architecture of mobile applications.

CO 3. Understand enterprise scale requirements of mobile applications.

CO 4. Design and develop mobile applications using one application development framework.

Course Outcomes	Program Outcomes													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDE0454														
CO 1														
CO 2			M				L							
CO 3		L				M								L
CO 4				M				H	H					

Course Content:

Unit 1: Introduction: Mobile operating system, Operating system structure, Constraints and Restrictions, Hardware configuration with mobile operating system, Features: Multitasking Scheduling, Memory Allocation, FileSystem Interface, Keypad Interface, I/O Interface, Protection and Security, Multimedia features.

Unit 2: J2ME Overview, J2ME Architecture and Development Environment, J2ME Best Practices and Patterns, High-Level Display, Low-Level Display, Record Management System

Unit 3: JDBC Objects: The Concept of JDBC, JDBC Driver Types, JDBC Packages, Overview of the JDBC Process, Database Connection, statement Objects, Result set, Transaction Processing, Metadata, Data Types, Exceptions. JDBC and Embedded SQL: Model Programs, Tables, Indexing, Inserting Data into Tables, Selecting Data from a Table, Metadata, Updating Tables, Deleting Data form a Table, Joining Tables, Calculating Data, Grouping and Ordering Data, Subqueries, VIEWs

Unit 4: Android: Introduction to Android, Architecture, memory management, communication protocols, application development methods, deployment, dealing with screen configurations and multiple screen sizes, Activity Class and its lifecycle, intents and permissions, Fragment Class, creating user interfaces, notifying users about important events, handling concurrency, acquiring data over the network, Mobile Cloud Computing with Android (MoCCA).

Text/Reference Books:

1. Jeff McWherter, Scott Gowell "Professional Mobile Application Development", John Wiley & Sons.
2. J2ME: The Complete Reference, James Keogh, Tata McGraw Hill.
3. Enterprise J2ME: Developing Mobile Java Applications – Michael Juntao Yuan, Pearson Education.
4. Beginning J2ME: From Novice to Professional, Third Edition, Sing Li, Jonathan B. Knudsen, A press,2005.
5. Jochen Schiller, "Mobile Communications", Addison-Wesley, 2nd edition, 2004.

Course Code	**ITDE0456**
Subject Name	**HIGH PERFORMANCE COMPUTING**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**None**

Course Objective:
At the end of the course, students will be able to understand the concepts of: Parallel Computing, New Processor Architectures, Power-Aware Computing and Communication, Advanced Topics on Computing.

Course Outcomes:

CO 1. To study parallel models of computation such as dataflow, and demand-driven computation
CO 2. To understand the need to achieve higher performance in modern computing systems.
CO 3. To apply various high performance computing approaches for a given problem.
CO 4. To evaluate the advanced topics on computing.

Course Outcomes	Program Outcomes													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDE0456														
CO 1	H	M				L		L					M	
CO 2			M	H	H					L	L		M	
CO 3		M		M	L				M			M	H	M
CO 4	H				H					M	L	H	H	H

Course Content:

Unit 1: Parallel Processing Concepts; Levels and model of parallelism: instruction, transaction, task, thread, memory, function, data flow models, demand-driven computation.

Unit 2: Parallel architectures: superscalar architectures, multi-core, multi-threaded, server and cloud; Fundamental design issues in HPC: Load balancing, scheduling, synchronization and resource management.

Unit 3: Operating systems for scalable HPC; Parallel languages and programming environments; Performance analysis of parallel algorithms; Fundamental limitations in HPC: bandwidth, latency and latency hiding techniques; Scalable storage systems: RAID, SSD cache, SAS, SAN; HPC based on cluster, cloud, and grid computing: economic model, infrastructure, platform, computation as service;

Unit 4: Accelerated HPC: architecture, programming and typical accelerated system; Power-aware HPC Design: computing and communication, processing, memory design, interconnect design, power management; Advanced topics: peta scale computing; big data processing, optics in HPC, quantum computers.

Text/Reference Books:

1. George S. Almasi and Alan Gottlieb, "Highly Parallel Computing", Addison Wesley Longman, 2/e, 1993.
2. Kai Hwang, "Advanced Computer Architecture: Parallelism, Scalability, Programmability", McGraw Hill 1993
3. David Culler Jaswinder Pal Singh, Morgan Kaufmann, "Parallel Computer Architecture: A hardware/Software Approach", 1999.
4. Kai Hwang, "Scalable Parallel Computing", McGraw Hill 1998.
5. "Principles and Practices on Interconnection Networks", by William James Dally and Brian Towles, Morgan Kauffman 2004.
6. GPU Gems 3 --- by Hubert Nguyen (Chapter 29 to Chapter 41)
7. Ananth Grama, Anshul Gupta, George Karypis, and Vipin Kumar, Introduction to Parallel Computing, 2nd edition, Addison-Welsey, 2003.
8. Petascale Computing: Algorithms and Applications, David A. Bader (Ed.), Chapman & Hall/CRC Computational Science Series, 2007.
9. Georg Hager, Gerhard Wellein, Introduction to High Performance Computing for Scientists and Engineers, Chapman & Hall / CRC Computational Science series, 2011.

DEPARTMENTAL ELECTIVES VI

S.No.	Subject Code	Subject Name	L	T	P	Credits
1	ITDE0462	Natural Language Processing	3	0	0	3
2	ITDE0464	Computer Vision	3	0	0	3
3	ITDE0466	Federated Learning	3	0	0	3

Course Code	**ITDE0462**
Subject Name	**NATURAL LANGUAGE PROCESSING**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**None**

Course Objective:

At the end of the course students will be able to learn how one could develop natural language understanding models from regularities in large corpora of natural language texts.

Course Outcomes:

CO 1. To understand the mathematical foundations, Probability theory with Linguistic essentials such as syntactic and semantic analysis of text.
CO 2. To analyze various natural language concepts and language modelling technique based on the structure of the language.
CO 3. To familiarity to linguistics and their application to part-of-speech tagging.
CO 4. To analyze and develop real-world applications based on natural language concepts.

Course Outcomes	**Program Outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDE0462														
CO 1			M			H		M		M			L	L
CO 2	M	H		M		H							M	L
CO 3			H		H			H	H					M
CO 4				M	M						M	M	H	H

Course Content:

Unit 1: Introduction: NLP introduction, origins of NLP, Language and Knowledge, The challenges of NLP, Different levels of NLP; Text Normalization: Basic pre-processing, Word and sentence segmentation, Lemmatization, Stemming, Morphology;

Unit 2: Language Models: n-gram models, smoothing techniques; Sequence Learning Tasks and Models, Language and Grammar, Processing Indian Languages, NLP applications, Some successful Early NLP systems, Information Retrieval Language Modeling: Introduction, Various Grammars- based language models, Statistical Language Model. Word Level Analysis: Introduction, Regular Expressions, Finite State Automata, Morphological Parsing, Spelling Error Detection and Correction, Words and Word Classes, Part-of-Speech Tagging.

Unit 3: Syntactic Parsing: Regular and Context-Free Languages, Context-Free Parsing, CKY Algorithm; Dependency Parsing: Dependency Grammar, Graph-based dependency parsing, Transition-based dependency parsing;

Unit 4: Vector-space Models: Word and Meanings, Distributional and Distributed Semantics, Lexical Semantic Analysis, GloVe, word2vec; Reference Resolution; Applications,.

Unit 5: Semantic Analysis: Introduction, Meaning Representation, Lexical Semantics, Ambiguity, Word Sense Disambiguation. Discourse Processing: Introduction, Cohesion, Reference Resolution, Discourse Coherence and Structure. Natural Language Generation: Introduction, Architecture of NLG Systems, Generation Tasks and Representations, Application of NLG.

Text/Reference Books:

1. Tanveer Siddiqui and U. S. Tiwary, "Natural Language Processing and Information Retrieval", Oxford Higher Education, 2008.
2. James Allen, "Natural Language Understanding", 2/e, Pearson Education, 1994.
3. Daniel Jurafsky, James H. Martin - "Speech and Language Processing", 3rd Edition, Prentice Hall, 2022
4. L.M. Ivansca and S. C. Shapiro, "Natural Language Processing and Language Representation", AAAI Press, 2000.
5. Akshar Bharti, Vineet Chaitanya and Rajeev Sangal, "NLP: A Paninian Perspective", Prentice Hall, New Delhi, 2004
6. J. Eisenstein, Introduction to Natural Language Processing, MIT Press, 2019.

Course Code	**ITDE0464**
Subject Name	**COMPUTER VISION**

Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objective:

To introduce students the major ideas, methods, and techniques of computer vision and pattern recognition.

Course Outcomes:

After the course completion, the student will be able to

CO 1. Understand key features of Computer Vision to analyse and interpret the visible world around us.

CO 2. Design and implement multi-dimensional signal processing, feature extraction, pattern analysis visual geometric modelling, and stochastic optimization.

CO 3. Apply the computer vision concepts to Biometrics, Medical diagnosis, document processing, mining of visual content, to surveillance, advanced rendering.

CO 4. Understand and apply the concepts of computer vision for implementation of detection techniques.

Course Outcomes	**Program Outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDE0464														
CO 1	L	H		M				L						
CO 2	L	M						M					L	
CO 3	L		M			H						L		
CO 4	L	M			M									

Course Content:

Unit 1: Introduction to Computer Vision: Overview and State-of-the-art, The Four Rs of Computer Vision, Geometry of Image Formation, Digital Image Formation and low-level processing, Fundamentals of Image Formation, Transformation: Orthogonal, Euclidean, Affine, Projective, etc; Fourier Transform, Convolution and Filtering, Image Enhancement, Restoration, Histogram Processing, Two View Geometry, Planar Scenes and Homography, Interest Point Detection, Depth estimation and Multi-camera views, Robust Correspondence Estimation, Perspective, Edge Detection.

Unit 2: Binocular Stereopsis: Camera and Epipolar Geometry; Image Filtering Rectification, DLT, RANSAC, Hough Transform, 3-D reconstruction framework; Autocalibration. apparel, Feature Extraction, Edges - Canny, LOG, DOG; Line detectors (Hough Transform), Corners - Harris and Hessian Affine, Orientation Histogram, SIFT, SURF, HOG, GLOH, Scale-Space Analysis- Image Pyramids and Gaussian derivative filters, Gabor Filters and DWT.

Unit 3: Recognition: Building blocks, Detectors and Descriptors, SIFT & Single Object Recognition, Optical Flow & Tracking, Introduction to Object Recognition and Bag-of-Words Models, Constellation model, Recognition: Objects, Scenes, Activities, Object classification and detection: a part-based discriminative model (Latent SVM), Objects in Scenes.

Unit 4: Light at Surfaces; Phong Model; Reflectance Map; Albedo estimation; Photometric Stereo; Use of Surface Smoothness Constraint; Shape from Texture, color, motion and edges, Face Detection, Deep Learning, Image Segmentation, Feature Tracking & Motion Layers.

Text/Reference Books:

1. Computer Vision: A Modern Approach, D. Forsyth and J. Ponce, Prentice Hall, 2nd ed., 2015, 2nd Edition.

2. Prince, Simon JD. Computer vision: models, learning, and inference. Cambridge University Press, 2012, 1st Edition.

3. Computer Vision: Algorithms and Applications, by Richard Szeliski, 2011 Edition, Springer.

4. Introductory Techniques for 3D Computer Vision, Emanuele Trucco and Alessandro Verri, Prentice Hall.1998, 1st Edition.

Course Code	**ITDE0466**
Subject Name	**FEDERATED LEARNING**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives:

1. To know the basics of federated learning.
2. To learn the applications of federated learning.
3. To know the privacy preserving deep learning.

Course Outcomes:

CO 1. Knowledge of basic concepts, architecture and applications of federated learning.
CO 2. Understanding of new research and application trends in federated learning.
CO 3. Ability to deploy the real-world federated learning projects.
CO 4. Hands-on experience in applying federated learning tools to solve privacy-preserving Artificial intelligence-based challenges.

Course Outcomes	**Program Outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITDE0466														
CO 1			M											
CO 2	L	M	L		M			M					M	
CO 3										L				
CO 4		L												H

Course Content:

Unit 1: Introduction to federated learning- Privacy Preserving-Distributed Machine Learning-Threats to federated learning-Data Valuation.
Horizontal Federated Learning- Vertical Federated Learning and Federated Transfer Learning.

Unit 2: Federated Optimization for Homogeneous Networks- Deep Networks from Decentralized Data- Federated Multi-task Learning-Personalized Federated Learning.
Federated Learning Applications: Recommendation in Health Care and Finance- Mobile Keyboard prediction- Learning of out-of-vocabulary words.

Unit 3: Adaptive personalized Federated Learning - Privacy Preserving Deep learning- Advances and open problems. Federated Learning techniques: FedAvg, FedSGD, FedProx, FedDANE
Federated Learning with Differential Privacy: Differential Privacy, Introduction to DP+ FedAvg with DP.

Unit 4: Federated Learning on Cloud: Setup the cloud environment, Implementing FedAvg on cloud with multiple machines

Text/Reference Books:

1. George I. Roussos and George Mastorakis, "Federated Learning: Strategies for Improving Communication Efficiency", Springer, 1st ed. 2020.
2. Xiang Li, Weiwei Yang, and Qiang Yang, "Advances in Federated Learning: Theory and Applications", Springer; 1st ed., 2021
3. Li Yang, Jie Xu, and Qingquan Zhang, “Federated Learning: Algorithms, Systems, and Applications”, Springer; 1st ed. 2021.
4. Kim-Kwang Raymond Choo, Ali Dehghantanha, Handbook of Big Data Privacy, Springer Nature Switzerland, 2020.
5. Qiang Yang, Yang Liu, Yong Cheng, Yan kang, Tianjian Chen, Han Yu, Federated Learning, Morgan & Claypool Publishers, 2019.

Generic Electives

GENERIC ELECTIVES I

S.No.	Subject Code	Subject Name	L	T	P	Credits
1	ITGE0351	Web Design Concepts	3	0	0	3
2	ITGE0353	Agile Software Engineering	3	0	0	3

Course Code	**ITGE0351**
Subject Name	**WEB DESIGN CONCEPTS**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives:

1. Introduction and brief history of world wide web (WWW).
2. Web essentials: HTML, XHTML, CSS.
3. Addressing web standards, audience requirements and principles of web page design.
4. Introduction of Web architecture, databases, jdbc.

Course Outcomes:

CO 1. Understand basic principles of web site design, considering the information architecture.
CO 2. Incorporate best practices in navigation, usability in website design
CO 3. Design of website adhering to current web standards (HTML, XML, CSS)
CO 4. Evaluating various scripting languages to create interactive components in web pages.

Course Outcomes	Program Outcomes													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITGE0351														
CO 1	H		M		M									
CO 2	M	M	H	M	M				H					
CO 3					H	M							M	
CO 4		H		H		H		H		M		M	M	

Course Content:

Unit 1: Introduction to HTML: HTML Common tags- List, Tables, images, forms, Frames; Cascading Style sheets;

Introduction to JavaScript: Scripts, Objects in Java Script, Dynamic HTML with Java Script XML: Document type definition, XML Schemas, Document Object model, Presenting XML, Using XML Processors: DOM and SAX

Unit 2: Java Beans: Introduction to Java Beans, Advantages of Java Beans, BDK Introspection, Using Bound properties, Bean Info Interface, Constrained properties Persistence, Customizes, Java Beans API, Introduction to EJB's

Unit 3: Web Servers and Servlets: Tomcat web server, Introduction to Servelets: Lifecycle of a Serverlet, JSDK, The Servelet API, The javax.servelet Package, Reading Servelet parameters, Reading Initialization parameters. The javax.servelet HTTP package, Handling Http Request & Responses, Using Cookies-Session Tracking, Security Issues,

Unit 4: Introduction to JSP: The Problem with Servelet. The Anatomy of a JSP Page, JSP Processing. JSP Application Design with MVC Setting Up and JSP Environment: Installing the Java Software Development Kit, Tomcat Server & Testing Tomcat

Unit 5: JSP Application Development: Generating Dynamic Content, Using Scripting Elements Implicit JSP Objects, Conditional Processing – Displaying Values Using an Expression to Set an Attribute, Declaring Variables and Methods Error Handling and Debugging Sharing Data Between JSP pages, Requests, and Users Passing Control and Date between Pages – Sharing Session and Application Data – Memory Usage Considerations
Database Access: Database Programming using JDBC, Studying Javax.sql.* package, Accessing a Database from a JSP Page, Application – Specific Database Actions, Deploying JAVA Beans in a JSP Page, Introduction to struts framework.

Text/Reference Books:

1. Chris Bates ,"Web Programming, building internet applications", 3rd Ed., WILEY Dreamtech, 2007.
2. Patrick Naughton and Herbert Schildt, "The complete Reference Java 2" 5th Ed., TMH, 2002.
3. Hans Bergsten , "java Server Pages", 2nd Ed., O'Reilly, 2002.
4. Dietel, Nieto, "Internet and World Wide Web – How to program", 2nd Ed., PHI, 2001.
5. Joel Sklar, "Web Warrier guide to web design technologies", 3rd Ed. Cengage Learning, 2005.

Course Code	**ITGE0353**
Subject Name	**AGILE SOFTWARE ENGINEERING**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives:

1. To explain how an iterative, incremental development process leads to faster delivery of more useful software
2. To discuss the essence of agile development methods
3. To explain the principles and practices of extreme programming
4. To explain the roles of prototyping in the software process

Course Outcomes:

CO 1. Describe two or more agile software development methodologies.
CO 2. Identify the benefits and pitfalls of transitioning to agile.
CO 3. Compare agile software development to traditional software development models.
CO 4. Apply agile practices such as test-driven development, standup meetings, and pair programming to their software engineering practices.

Course Outcomes	Program outcomes													
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO1 0	PO1 1	PO1 2	PSO 1	PSO 2
ITGE0353														
CO 1	H						M		H				H	
CO 2	M	M	H					H		L				M
CO 3	M	H	M					M	M	M		H		
CO 4	M	H			M					M	H	M	M	H

Course Contents:

Unit 1: Overview of Software Engineering: Process, Project, Product, Method, Tool, Modern Life cycle, Traditional development approaches

Unit 2: Advanced Process models: V-Model, Component based development model, Agile Development Model, Unified Process Model, Extreme Programming, Feature Driven development, Lean Software Development, Service Oriented Architecture, Aspect Oriented Development

Unit 3: Agile Project Management: Agile Scrum Framework, Project Planning, Scheduling, Agile Estimation, Iterative Planning, Roles

Software Specification: New paradigms in software specification and design, Agile Specification, Short review of UML.

Unit 4: Design Engineering: Software architecture, Object-oriented Design, Software Patterns, Pattern-oriented Design, Component-oriented design. Software Frameworks, Agile Design

Unit 5: Agile Testing and Test Driven Development: The Agile lifecycle and its impact on testing, Test-Driven Development (TDD), xUnit framework and tools for TDD, Testing user stories - acceptance tests and scenarios, Planning and managing testing cycle, Exploratory testing, Risk based testing, Regression tests, Test Automation, Tools to support the Agile tester.

Text/Reference Books:

1. Roger S. Pressman - "Software Engineering: A Practitioner's Approach", 9th Edition, McGraw-Hill Education, 2019.
2. Ian Sommerville - "Software Engineering", 10th Edition, Pearson, 2015.
3. Agile Software Development with Scrum By Ken Schawber, Mike Beedle, Pearson, 2008
4. Robert C. Martin - "Clean Agile: Back to Basics", Prentice Hall, 2020.

GENERIC ELECTIVES II

S.No.	Subject Code	Subject Name	L	T	P	Credits
1	ITGE0352	Fundamentals of Data Analytics	3	0	0	3
2	ITGE0354	Fundamentals of Cloud Computing	3	0	0	3

Course Code	**ITGE0352**
Subject Name	**FUNDAMENTALS OF DATA ANALYTICS**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives:

1. Discuss the overall process of how data analytics is applied
2. Discuss how data analytics can be used to better address and identify risks
3. Demonstrate the power of data analytics using case studies

Course Outcomes:

CO 1. Understand what Big Data is and why classical data analysis techniques are no longer adequate.

CO 2. Learn the benefits that Big Data can offer to businesses and organizations.

CO 3. Analyze conceptually how Big Data is stored.

CO 4. Apply how Big Data can be analyzed to extract knowledge.

Course Outcomes	Program Outcomes													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITGE0352														

CO 1	M	H	L				L				H	H		
CO 2		L	M			M			H		M			
CO 3	M	L					L				M	H		
CO 4	H	M	H		L				L		M			M

Course Content:

Unit 1: Introduction: Big Data Overview, Challenges with Big Data, Traditional Business Intelligence (BI) versus Big Data.

Big Data Analytics: The rising and importance of data sciences, Classification of Analytics, Data Sciences, Terminologies in Big Data Environment (In –Memory Analytics, In-Database Processing, Symmetric Multiprocessor system, Massively parallel processing, Parallel and distributed system, Shared nothing architecture, CAP Theorem), Analytics tools.

Unit 2: Hadoop Overview: Introduction to Hadoop, RDBMS vs Hadoop, key aspects of hadoop, hadoop components, hadoop conceptual layer, high level architecture of hadoop.

Hadoop Architecture: Hadoop architecture, Hadoop ecosystem components, Hadoop storage: HDFS, Hadoop processing, Map Reduce Framework, Hadoop server roles.

Unit 3: Hadoop big data technology landscape: NoSQL, Types of NoSQL database, Advantages, New SQL, Comparison of SQL, NoSQL and NewSQL.

Unit 4: Overview of Data Analytics- Theory and Methods: Measures and evaluation, Supervised Learning, Linear and Logistic Regression, Decision trees, Unstructured data analytics.

Text/Reference Books:

1. Seema Acharya, Subhashini Chellappan - "Big Data and Analytics", Wiley, 1st Edition, 2021.
2. Tom White - "Hadoop: The Definitive Guide: Storage and Analysis at Internet Scale", 4th Edition, O'Reilly, 2015.
3. David Loshin - "Big Data Analytics: From Strategic Planning to Enterprise Integration with Tools, Techniques, NoSQL, and Graph", Morgan Kaufmann, 1st Edition, 2021.

Course Code	**ITGE0354**
Subject Name	**FUNDAMENTALS OF CLOUD COMPUTING**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives:

1. To study how to use Cloud Services.
2. To implement Virtualization
3. To implement Task Scheduling algorithms, Map-Reduce concept to applications.
4. Broadly educate to know the impact of engineering on legal and societal issues involved

Course Outcomes:

CO 1. To understand and apply the knowledge of systems protocols and mechanisms to support Cloud computing.

CO 2. To analyze and implement the concepts of Cloud Computing.

CO 3. To deploy applications on cloud middleware.

CO 4. To apply and resolve security issues in various applications.

Course Outcomes	**Program Outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITGE0354														
CO 1	L	H		M										

CO 2	L	M	M					M						
CO 3	L	M	M			H				H			H	
CO 4	L	M	M		M								H	

Course Content:

Unit 1: Cloud Computing, History of Cloud Computing, Cloud Architecture, Cloud Storage, Why Cloud Computing Matters, Advantages of Cloud Computing, Disadvantages of Cloud Computing, Companies in the Cloud Today, Cloud Services

Unit 2: Web-Based Application, Pros and Cons of Cloud Service Development, Types of Cloud Service Development, Software as a Service, Platform as a Service, Web Services, On-Demand Computing, Discovering Cloud Services Development Services and Tools, Amazon Ec2, Google App Engine, IBM Clouds

Unit 3: Centralizing Email Communications, Collaborating on Schedules, Collaborating on To-Do Lists, Collaborating Contact Lists, Cloud Computing for the Community, Collaborating on Group Projects and Events, Cloud Computing for the Corporation

Unit 4: Collaborating on Calendars, Schedules and Task Management, Exploring Online Scheduling Applications, Exploring Online Planning and Task Management, Collaborating on Event Management, Collaborating on Contact Management, Collaborating on Project Management, Collaborating on Word Processing - Collaborating on Databases, Storing and Sharing Files

Unit 5: Collaborating via Web-Based Communication Tools, Evaluating Web Mail Services, Evaluating Web Conference Tools, Collaborating via Social Networks and Groupware, Collaborating via Blogs and Wikis

Text/Reference Books:

1. Michael Miller - "Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate Online", Que Publishing, 1st Edition, August 2008.
2. Haley Beard - "Cloud Computing Best Practices for Managing and Measuring Processes for On-demand Computing, Applications and Data Centers in the Cloud with SLAs", Emereo Pty Ltd, 1st Edition, July 2008.
3. Thomas Erl, Ricardo Puttini, Zaigham Mahmood - "Cloud Computing: Concepts, Technology & Architecture", Prentice Hall, 2013.
4. Kailash Jayaswal, Jagannath Kallakurchi - "Cloud Computing: Fundamentals and Overview", Wiley India, 2021.
5. Rajkumar Buyya, Satish Narayana Srirama - "Fog and Edge Computing: Principles and Paradigms", Wiley, 2019.

GENERIC ELECTIVES III

S.No.	Subject Code	Subject Name	L	T	P	Credits
1	ITGE0451	Machine Learning	3	0	0	3
2	ITGE0453	Fundamental of Artificial Intelligence	3	0	0	3

Course Code	**ITGE0451**
Subject Name	**MACHINE LEARNING**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives:

This course will serve as a comprehensive introduction to various topics in machine learning. At the end of the course the students should be able to design and implement machine learning solutions to classification, regression, and clustering problems; and be able to evaluate and interpret the results of the algorithms.

Course Outcomes:

CO 1. Understanding about the basic principles, techniques and applications of machine learning.
CO 2. Broad understanding of machine learning algorithms and their use in data-driven knowledge discovery and program synthesis.
CO 3. Knowledge of the strengths and weaknesses of different machine learning algorithms (relative to the characteristics of the application domain).
CO 4. The ability to adapt or combine some of the key elements of existing machine learning algorithms to design new algorithms as needed.

Course Outcomes	**Program Outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITGE0451														
CO 1		M	M	M	M	H								
CO 2		M	H	M	M	H	H	M						
CO 3		H	M	H	M									
CO 4		H	H	M	M	M	M							M

Course Content:

Unit 1: Introduction Well-Posed Learning Problems, Designing a Leaning System, Perspectives and Issues in Machine Learning. Concept Learning and the General-to-Specific Ordering: Introduction, A Concept Learning Task, Concept Learning as Search, FIND-S: Finding a Maximally Specific Hypothesis, Version Spaces and the CANDIDATE-ELIMINATION Algorithm.

Unit 2: Decision Tree Learning: Introduction, Decision Tree Representation, Appropriate problem for Decision tree Learning, The Basic Decision Tree Learning Algorithm, Hypothesis Space Search in Decision Tree Leaning, Inductive Bias in Decision Tree Leaning, Issues in Decision Tree Leaning

Unit 3: Artificial Neural Networks: Introduction, Natural Network Representations, Appropriate Problems for Neural Network Learning, Perceptions, Multilayer Network and the BACKPROPAGATION Algorithm Bayesian Learning: Introduction, Bayes Theorem, Bayes Theorem and Concept Learning, Bayes Optimal Classifier, Native Bayes Classifier, An Example: Learning to Classify Text. Instance- Based Learning: Introduction, K-NEAREST NEIGHBOUR Learning, Distance Weighted NEAREST NEIGHBOUR Algorithm.

Unit 4: Genetic Algorithms: Motivation, Genetic Algorithms, Hypothesis Space Search, Genetic Programming, Parallelizing Genetic Algorithms. Learning Sets of Rules: Introduction, Sequential Covering Algorithms, Learning Rule Sets: Summary, Learning First-Order Rules, Learning Sets of First-Order Rules: FOIL, Induction as Inverted Deduction, Inverted Resolution.
Support Vector Machine: Maximum margin linear separators, Quadratic Programming Solution to finding maximum margin separators, Kernels for learning non-linear functions.

Text/Reference Books:

1. Tom Mitchell, “Machine Learning”, McGraw-Hill, 1997.
2. E. Alpaydin, “Machine Learning”, MIT Press, 2010.
3. K. Murphy, “Machine Learning: A Probabilistic Perspective”, MIT Press, 2012.
4. C. Bishop, “Pattern Recognition and Machine Learning, Springer”, 2006.

Course Code	**ITGE0453**
Subject Name	**FUNDAMENTALS OF ARTIFICIAL INTELLIGENCE**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives:

1. To provide a strong foundation of fundamental concepts in Artificial Intelligence
2. To provide a basic exposition to the goals and methods of Artificial Intelligence
3. To enable the student to apply these techniques in applications which involve perception, reasoning and learning

Course Outcomes:

CO 1. Identify problems that are amenable to solution by AI methods, and which AI methods may be suited to solving a given problem.
CO 2. Formalize a given problem in the language/framework of different AI methods
CO 3. Describe basic AI algorithms (e.g., standard search algorithms or resolution).
CO 4. Design and carry out an empirical evaluation of different algorithms on a problem formalization.

Course Outcomes	**Program Outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITGE0453														
CO 1		H	M	L	L									
CO 2		M	H	H									L	
CO 3		M	M	L	M									
CO 4		H	H	H										M

Course Content:

Unit 1: Introduction: Introduction to AI: Definitions, Historical foundations, Basic Elements of AI, Characteristics of intelligent algorithm, AI application Areas, Introduction to genetic algorithm, Evolutionary searching algorithm.

Unit 2: Problem solving: State space search; Production systems, search space control: depth-first, breadth-first search, heuristic search - Hill climbing, best-first search, branch and bound. Problem Reduction, Constraint Satisfaction End, Means-End Analysis

Unit 3: Handling uncertainty: Non-Monotonic Reasoning, Probabilistic reasoning, use of certainty factors, Fuzzy logic.
Knowledge Based Systems: Proportional Logic, FOPL, Clausal Form, Resolution & Unification. Knowledge representation, acquisition, organisation & Manipulation, Semantic nets, Frames, Conceptual Dependency, Scripts & CYC.

Unit 4: Machine Learning: Concept of learning, Concept creation, learning automation, supervised and
Unsupervised Learning, learning tasks & learning strategies, single layer & multiplayer Perceptions, Back propagation, learning by inductions, Competitive Learning, Hebbian Coincidence Learning, Attractor Networks Samuel's checkers algorithm. Hopfield nets, Adaptive

resonance theory

Unit 5: Expert Systems: Need and justification for expert systems, Basic Components & architecture of Expert systems, ES-Shells, Representing & Using Domain Knowledge, Knowledge acquisition in expert Systems. Case studies: MYCIN, RI.

Text/Reference Books:

1. Elaine Rich, Kevin Knight - "Artificial Intelligence", 3rd Edition, McGraw-Hill, 2010.
2. George F. Luger - "Artificial Intelligence: Structures and Strategies for Complex Problem Solving", 6th Edition, Pearson, 2008.
3. Stuart Russell, Peter Norvig - "Artificial Intelligence: A Modern Approach", 4th Edition, Pearson, 2020.
4. Dan W. Patterson - "Introduction to Artificial Intelligence and Expert Systems", Prentice Hall, 1990.
5. Eugene Charniak, Drew McDermott - "Introduction to Artificial Intelligence", Addison-Wesley, 1985.
6. Nils J. Nilsson - "Principles of Artificial Intelligence", Narosa Publishing, 1982.

GENERIC ELECTIVES IV

S.No.	Subject Code	Subject Name	L	T	P	Credits
1	ITGE0452	Mobile Application Development Concepts	3	0	0	3
2	ITGE0454	Low Code/No Code Technology	3	0	0	3

Course Code	**ITGE0452**
Subject Name	**MOBILE APPLICATION DEVELOPMENT CONCEPTS**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives:

1. Introduction and characteristics of mobile applications.
2. Application models of mobile application frameworks.
3. Managing application data and User-interface design for mobile applications.
4. Integrating networking, the OS and hardware into mobile-applications.
5. Addressing enterprise requirements in mobile applications – performance, scalability, modifiability, availability and security.

Course Outcomes:

CO 1. Understand technology and business trends impacting mobile applications
CO 2. Be competent with the characterization and architecture of mobile applications.
CO 3. Understand enterprise scale requirements of mobile applications.
CO 4. Design and develop mobile applications using one application development framework.

Course Outcomes	Program Outcomes													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITGE0452														
CO 1			M				L							
CO 2		L				M								
CO 3				M				H	H					
CO 4		L					L						H	

Course Content:

Unit 1: Introduction: Mobile operating system, Operating system structure, Constraints and Restrictions, Hardware configuration with mobile operating system, Features: Multitasking Scheduling, Memory Allocation, FileSystem Interface, Keypad Interface, I/O Interface, Protection and Security, Multimedia features.

Unit 2: J2ME Overview: Java 2 Micro Edition and the World of Java, Inside J2ME, J2ME and Wireless Devices Small Computing Technology: Wireless Technology, Radio Data Networks, Microwave Technology, Mobile Radio Networks, Messaging, Personal Digital Assistants. J2ME Architecture and Development Environment, J2ME Best Practices and Patterns.

Unit 3: High-Level Display: Screens: Screen Class, Alert Class, Form Class, Item Class, List Class, TextBox Class, Ticker Class. Low-Level Display: Canvas: The Canvas, User Interactions, Graphics, Clipping Regions, Animation. Record Management System: Record Storage, Writing and Reading Records, Record Enumeration, Sorting Records, Searching Records, Record Listener.

Unit 4: JDBC Objects: The Concept of JDBC, JDBC Driver Types, JDBC Packages, Overview of the JDBC Process, Database Connection, statement Objects, Result set, Transaction Processing, Metadata, Data Types, Exceptions.

JDBC and Embedded SQL: Model Programs, Tables, Indexing, Inserting Data into Tables, Selecting Data from a Table, Metadata, Updating Tables, Deleting Data form a Table, Joining Tables, Calculating Data, Grouping and Ordering Data, Subqueries, VIEWs

Unit 5: Android: Introduction to Android, Architecture, memory management, communication protocols, application development methods, deployment, dealing with screen configurations and multiple screen sizes, Activity Class and its lifecycle, intents and permissions, Fragment Class, creating user interfaces, notifying users about important events, handling concurrency, acquiring data over the network, Mobile Cloud Computing with Android (MoCCA).

Text/Reference Books:

1. C. Firza Afreen, Mobile Applications Development. N.p.: Book Rivers, 2021.
2. Burd, Barry., Mueller, John Paul. Android Application Development All-in-One For Dummies. United Kingdom: Wiley, 2020.
3. Jeff McWherter, Scott Gowell “Professional Mobile Application Development”, John Wiley & Sons, 2012.
4. James Keogh, J2ME: The Complete Reference, Tata McGraw Hill, 2017.
5. Michael Juntao Yuan, Enterprise J2ME: Developing Mobile Java Applications, Pearson Education, 2003.
6. Sing Li, Jonathan B. Knudsen, Beginning J2ME: From Novice to Professional, Third Edition, A Press, 2005.
7. Jochen Schiller, “Mobile Communications”, Addison-Wesley, 2nd edition, 2004.

Course Code	**ITGE0454**
Subject Name	**LOW CODE/ NO CODE TECHNOLOGY**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objective:
To embark students with the no platform reliability concept and enable them to quickly build applications and automation concepts without any knowledge of how coding works to address their own development needs.

Course Outcomes:
On successful completion of this course, the students will be able to:

CO 1. Understand Low-Code/ No-Code platform.

CO 2. Analyzing the difference between Low Code and No Code.

CO 3. Understand Citizen Developers.

CO 4. Understanding the development of Web and Mobile Applications using Low-Code/ No-Code

Course Outcomes	**Program Outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITGE0454														
CO 1	H										M		M	
CO 2	M			M							M	M		
CO 3	M	H		M		H	L	M			M	M	M	
CO 4			H	M	H	M	L	M	M	M	M	H	M	H

Course Content:

Unit 1: Workplace Tech: The struggle is real, Isolating cause and effect, Who's to blame for the status Quo?, The pandemic validated the importance of workplace Tech, Understanding the dual nature of contemporary IT, Retention issues and remote work. How organizations launch new workplace technologies, Development methods.

Unit 2: Why Low-code/No-code changes everything: Building blocks and precursors, Distinguishing between No-Code and Low-Code tools, The major characteristics of Low-Code/ No-Code tools, The burgeoning industry, Major Low-Code/ No-Code subcategories, Contextualization Low-Code/ No-Code.

Unit 3: Introducing the citizen developer: History and definition, The rise of citizen developer, Common attributes of citizen developers, The benefits of citizen developer, IT benefits, Organizational benefits, Individuals and Team benefits.

Unit 4: Citizen development in Action: The municipality of rotterdam, A bubbly outcome for a budding entrepreneur, Low-Code/ No-Code powers up synergis education, Low-code/No-code transforms a family business, The skunworks approach, the best-of-breed approach, the laissez faire approach, the wait-and-see approach, the universal ban

Unit 5: How to evaluate Low-code/No-code tools and learn new one: Evaluating existing tools, Learning new Low-Code/ No-Code tools, Planning and gathering requirements, Design and development, Testing, Launch, Support, Maintenance, and Documentation, Retirement, Low-Code/ No-Code tools and their offspring. Where we go from here.

Text/Reference Books:

1. Phil Simon, Low-Code/No-Code: Citizen Developers and the Surprising Future of Business Applications, Racket Publishing, 2022
2. Kenneweg, Bryan., Kasam, Imran., McMullen, Micah., Guido, Michael. Building Low-Code Applications with Mendix: Discover Best Practices and Expert Techniques to Simplify Enterprise Web Development. India: Packt Publishing, 2021.
3. Murru, Enrico. Hands-On Low-Code Application Development with Salesforce: Build Customized CRM Applications that Solve Business Challenges in Just a Few Clicks. United Kingdom: Packt Publishing, 2020.

SCHEME AND DETAILED SYLLABI FOR

MINOR DEGREE

B. TECH

IN

INFORMATION TECHNOLOGY

Effective from 2023 Batch onwards

Dr. B R AMBEDKAR NATIONAL INSTITUTE OF TECHNOLOGY

JALANDHAR – 144008

Total credit 20

Minor Degree Courses

Semester- III							
S.No.	**Course Code**	**Courses**	**L**	**T**	**P**	**Credit**	**Offerings**
1.	ITMI0201	Database Management Systems	3	0	0	3	Autumn (3rd)
2.	ITMI0231	Database Management Systems Lab	0	0	2	1	Autumn (3rd)
Semester- IV							
3.	ITMI0202	Algorithms and Complexity	3	0	0	3	Spring (4th)
4.	ITMI0232	Algorithms and Complexity Lab	0	0	2	1	Spring (4th)
Semester- V							
5.	ITMI030X	Elective-I	3	0	0	3	Autumn (5th)
Semester- VI							
6.	ITMI0302	Software Engineering	3	0	0	3	Spring (6th)
Semester- VII							
7.	ITMI040X	Elective-II	3	0	0	3	Autumn (7th)
Semester- VIII							
8.	ITMI040X	Elective-III	3	0	0	3	Spring (8th)
Total Credit			**18**	**0**	**2**	**20**	

List of Elective-I (Fifth Semester)

S.No.	Course Code	Courses	L	T	P	Credit
1.	ITMI0301	Web Technologies	3	0	0	3
2.	ITMI0303	Robot Process Automation	3	0	0	3
3.	ITMI0305	Data Science Programming	3	0	0	3

List of Elective-II (Seventh Semester)

S.No.	Course Code	Courses	L	T	P	Credit
1.	ITMI0401	Java Programming Concepts	3	0	0	3
2.	ITMI0403	Introduction to Data Analytics	3	0	0	3
3.	ITMI0405	Prompt Engineering	3	0	0	3

List of Elective-III (Eighth Semester)

S.No.	Course Code	Courses	L	T	P	Credit
1.	ITMI0402	Business Intelligence	3	0	0	3
2.	ITMI0404	Wireless Communication for Beyond 5G Networks and IoT	3	0	0	3
3.	ITMI0406	Soft Computing Techniques	3	0	0	3

Detailed Syllabus

Third Semester

Course Code:	**ITMI0201**
Subject Name	**Database Management Systems**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives

1. To learn data models, conceptualize and depict a database system using ER diagram.
2. To understand the internal storage structures in a physical DB design.
3. To know the fundamental concepts of transaction processing techniques.
4. To study information systems, types of information systems and advanced concepts.

Course Outcomes

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

CO 1. Explain the fundamentals of DBMS and build conceptual models of real-world problems.

CO 2. Design and develop a database schema for real world problems, construct complex SQL queries and define various Normal Forms.

CO 3. Illustrate transaction processing, concurrency control and recovery techniques.

CO 4. Understand advanced concepts in Information Systems.

CO 5. Describe the evolution of Information Systems, Executive's role, Characteristics and capabilities of Executive Support Systems.

Course Outcomes	**Program outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITMI0201														
CO 1		M			L							H		
CO 2	L		M					M					M	
CO 3					M				L				M	
CO 4		L											H	
CO 5	H		M	H									M	

Course Content

Unit 1: INTRODUCTION & RELATIONAL MODEL

Purpose of Database System, Views of data, Data models, Database management system, Three-schema architecture of DBMS, Components of DBMS. E/R Model, Conceptual data modeling, Entities, Entity types, Attributes, Relationships, Relationship types, E/R diagram notation, examples. Relational Data Model, Keys, Referential integrity and Foreign keys

Unit 2: DATABASE DESIGN

SQL - Introduction, Data definition in SQL, Table, Key and foreign key definitions, Update behaviors, Querying in SQL.Dependencies and Normal forms, Dependency theory, Functional dependencies, Armstrong's axioms for FD's, Definitions of 1NF, 2NF, 3NF and BCNF, Decompositions and Desirable properties of them, Algorithms for 3NF and BCNF normalization, 4NF, and 5NF.

Unit 3: TRANSACTIONS

Transaction processing and Error recovery, Concepts of transaction processing, ACID properties, Concurrency control, Locking based protocols for CC, Error recovery and logging, Undo, Redo, Undo-redo logging and Recovery methods.

Unit 4: INFORMATION SYSTEMS

Introduction, Collaboration, Communication, Enterprise Decision support systems, and Knowledge Management, Components of Information Systems, Types of Information Systems, Group Support Systems. Enterprise Information Systems, Management Information Systems.

Unit 5: ADVANCED CONCEPTS

Integration, Impacts, and the Future of Management-Support Systems, Knowledge-Based System, Knowledge Acquisition, Representation, and Reasoning.

Recommended books:

1. Silberschatz, A., Korth, Henry F., and Sudharshan, S., “Database System Concepts”, 5th Edition, Tata McGraw Hill, 2016.
2. Elmasri, Ramez and Navathe, Shamkant B., “Fundamentals of Database Systems”, 7th Edition, Pearson, 2015.
3. George M. Marakas, “Decision Support Systems in 21st century”, 2nd edition PHI.
4. Efraim Turban, Jay E. Aronson, Ting-Peng Liang, “Decision Support Systems and Intelligent Systems”, 6th edition PHI.

Course Code:	**ITMI0231**
Subject Name	**Database Management Systems Lab**
Contact hours/Credit Scheme (L-T-P-C)	**0-0-2-1**
Pre-requisites	**NONE**

Course Objectives

1. To understand the role of databases and database management systems in managing organizational data and information.
2. To understand the techniques used for data fragmentation, replication and allocation during the distributed database design process.
3. To Perceive the building blocks and design of information systems

Course Outcomes:

At the end of the course the students are able to:

CO 1. Apply the basic concepts of Database Systems and Applications.
CO 2. Use the basics of SQL and construct queries using SQL in database creation and interaction.
CO 3. Design a commercial relational database system (Oracle, MySQL) by writing SQL using the system.
CO 4. Analyze and Select storage and recovery techniques of database system.

Course Outcomes	**Program outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITMI0231														
CO 1		M			L							H		
CO 2	L		M					M					M	
CO 3					M				L				M	
CO 4		L											H	
CO 5	H		M	H									M	

LIST OF PRACTICALS

1. Implementation of DDL commands of SQL with suitable examples • Create table • Alter table • Drop Table.
2. Implementation of DML commands of SQL with suitable examples • Insert • Update • Delete
3. Implementation of different types of function with suitable examples • Number function • Aggregate Function • Character Function • Conversion Function • Date Function
4. Implementation of different types of operators in SQL • Arithmetic Operators • Logical Operators • Comparison Operator • Special Operator • Set Operation
5. Implementation of different types of Joins • Inner Join • Outer Join • Natural Join etc.
6. Study and Implementation of • Group By & having clause • Order by clause • Indexing
7. Study & Implementation of • Sub queries • Views
8. Study & Implementation of different types of constraints.
9. Study & Implementation of Database Backup & Recovery commands. Study & Implementation of Rollback, Commit, Savepoint.
10. • Creating Database /Table Space • Managing Users: Create User, Delete User • Managing roles:- Grant, Revoke.
11. Study & Implementation of PL/SQL.
12. Study & Implementation of SQL Triggers.
13. Mini-projects implementation in RDBMS environment.

This is only the suggested list of Practicals. Instructors may frame additional Practicals relevant to the course contents.

Fourth Semester

Course Code:	**ITMI0202**
Subject Name	**Algorithms and Complexity**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives

1. To impart the basic concepts of data structures and algorithms.
2. To understand writing algorithms and step by step approach in solving problems with the help of fundamental data structures.
3. To understand concepts about searching and sorting techniques.
4. To be familiar with basic techniques of algorithm analysis.
5. To learn and implement various data structures and algorithms.

Course Outcomes

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

CO 1. Explain the fundamentals of data structures and algorithms.

CO 2. Build data structures for a given problem.

CO 3. Illustrate applications and use of tree data structures using arrays and linked lists.

CO 4. Compare algorithms for tree and graph data structures.

CO 5. Compare the basic algorithmic techniques and choose a suitable solution for a given problem.

Course Outcomes	**Program outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITMI0202										H				
CO 1	M	M			M								H	H
CO 2	L	M		H									H	H
CO 3	H	H		H					L					
CO 4	H	M		H					H	H				
CO 5	H		H						H	H				

Course Content

Unit 1: Introduction

Notations and Analysis, Complexity and its Types, Storage structures for arrays, sparse matrices, structures and arrays of structures, Singly linked lists, Linked stacks and queues, operations on Polynomials, Doubly Linked Lists, Circularly Linked Lists, Operations on linked lists: Insertion, deletion and traversal.

Unit 2: Construction of Algorithms

Principles for construction of algorithms: Decomposition, greedy algorithms, dynamic programming, local and total search. Algorithm analysis. Approximation, algorithms and heuristics. Selected applications to sets, graphs, arithmetic, and geometry. Implementation of algorithms.

Unit 3: Graphs and Trees

Graphs: Flavors of Graphs, Data Structures for Graphs, traversing a Graph, Breadth First Search and its applications, Depth First Search and its applications, Depth First search on Directed Graphs, Directed and undirected graphs.

Hash tables and heaps; balanced trees and randomised data structures. Use and implementation of data structures.

Unit 4: Sorting and Searching

Heap Sort, Merge Sort, Sorting by Divide and Conquer, Quick sort: Sorting by Randomization, Distribution Sort: Sorting by Bucketing, Linear and Binary Search.

Unit 5: Advanced Concepts

Computability and complexity: Reduction. Complexity classes P (polynomial time) and NP (non-deterministic polynomial time). NP-complete problems. Undecidable problems. Coping with intractable problems.

Recommended books:

1. Kruse, Tondo and Leung, "Data Structures and Program Design in C edition", Prentice-Hall, 1997.
2. Skiena Steven S., "The Algorithm Design Manual" 2nd edition, Springer, 2008.
3. Lipschutz Seymour, "Data Structures", McGraw Hill revised first edition, 2014.
4. Cormen, T. H., Leiserson, C. E., & Rivest, R. L. (1990), Introduction to Algorithms, MIT Press, McGraw-Hill.

Course Code:	**ITMI0232**
Subject Name	**Algorithms and Complexity Lab**
Contact hours/Credit Scheme (L-T-P-C)	**0-0-2-1**
Pre-requisites	**NONE**

Course Objectives

1. Understand the use of arrays, records, linked structures, stacks, queues, trees, and graphs.
2. Understand how the choice of data structures can lead to efficient implementations of algorithms.
3. Familiarize how certain applications can benefit from the choice of data structures.
4. Apply and implement the learned algorithm for problem solving.

Course Outcomes

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

CO 1. Identify the problem given and design the algorithm using various algorithm design techniques.

CO 2. Implement various algorithms in a high level language.

CO 3. Analyze the performance of various algorithms.

CO 4. Compare the performance of different algorithms for same problem

Course Outcomes	Program outcomes													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITMI0232														
CO1	M	M			L	L				M	L		M	
CO2		L	M		H					M				M
CO3		M	H	M						L	M	M		H

LIST OF PRACTICALS

1. Implementation of various operations on arrays like Print the elements of an array in reverse order, delete an element from an array at a given index, find the smallest element in an array, calculate the sum of elements in an array etc.
2. Implementation of linear and binary search technique in an array of n values.
3. Implementation of various operations on linked list: Creation, Insertion, Deletion, Reverse etc.
4. Implementation of various operations on doubly linked list: Creation, Insertion, Deletion etc.
5. Implementation of stack data structure using array of n values and perform push () and pop () operation.
6. Implementation of stack operations to convert a given infix expression into its equivalent postfix expression. Implement the stack using an array.
7. Implementation of queue data structure using array and linked list with the basic functions of Create(), IsEmpty(), Insert(), Delete() and IsFull().
8. Implementation of circular queue using an array with the following operations: a) Insert a new element b) Delete a given element c) Display the content of queue
9. Implementation of a double ended queue using i) array and ii) doubly linked list respectively.
10. Implementation of sparse matrix and perform various operations like search the sparse matrix for an element specified by the user. Print the result of the search appropriately. Use the triple <row, column, value> to represent an element in the sparse matrix.
11. Implementations of Binary Tree menu driven program :
    a. Traverse the tree using all the methods i.e. inorder, preorder & postorder to display the elements in the tree

b. Copying tree
c. Counting the number of nodes in the tree d. Counting only leaf nodes in the tree.

12. Implementations of BST program with at least following operations.
    a) To construct a binary search tree of n values.
    b) To insert new elements in BST
    c) To delete elements in BST at various positions
13. Implementation of the following sorting methods to arrange a list of n values in ascending order: a) Insertion sort b) Merge sort c) Selection sort d) Bubble sort e) Quick sort f) Heap Sort
14. Implementation of the graph traversal algorithms: a) Depth first traversal b) Breadth first traversal

This is only the suggested list of Practicals. Instructors may frame additional Practicals relevant to the course contents.

List of Electives for Fifth Semester

S.No.	Course Code	Courses	L	T	P	Credit
1.	ITMI0301	Web Technologies	3	0	0	3
2.	ITMI0303	Robot Process Automation	3	0	0	3
3.	ITMI0305	Data Science Programming	3	0	0	3

Course Code	**ITMI0301**
Course Name	**Web Technologies (Elective)**
Contact Hours/ Credit Scheme	**L-T-P-C: 3-0-0-3**

Course Objectives

1. Introduction and brief history of world wide web (WWW).
2. Web essentials: HTML, XHTML, CSS.
3. Addressing web standards, audience requirements and principles of web page design.

Course Outcomes:

CO 1. To understand and compare the fundamentals of Web hosting and domain name services.
CO 2. To understand various non-browser specific web design principles.
CO 3. To understand the need and be able to develop HTML/XHTML and CSS pages with valid structure as well as content.
CO 4. To understand and be able to develop JavaScript/jQuery code to access the DOM structure of web document and object properties.
CO 5. To develop dynamic web pages with usage of server-side scripting PHP and MySQL.

Course Outcomes	**Program outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITMI0301														
CO 1	H		M		M									
CO 2	M	M	H	M	M				H				H	
CO 3					H	M								
CO 4		H		H		H		H		M		M	H	

Course Content

Unit 1: Basic design and implementation of websites:
Introduction to: the evolution of Web, Internet applications, TCP/IP, FTP, Telnet, HTTP, Important components of the Web: Firewalls, Routers, Browser, Web search engines, Web servers, Application servers.

Unit 2: Discussion of different navigation and organizational strategies:
Basic web navigation, role of website navigation, categories of navigation: Structural, Associative, Utility.

Unit-3: Client-side technologies: HTML5, CSS, JavaScript, JSON, and JQuery. XML: Document type definition, XML Schemas, Document Object model, Presenting XML, Using XML Processors: DOM and SAX

Unit 4: Server-side technologies: PHP, Declaring Variables, Data types, Operators, Control structures, Functions, Reading data from web form controls like text buttons, radio buttons, list, etc., Handling File Uploads, Handling Sessions & Cookies.

Unit 5: Back-end data management, Security issues, Emerging technologies.

Recommended Books:

1. "Web Technology: Theory and Practice", M. Srinivasan, Pearson India.
2. "Web Technologies", Uttam K Roy, Oxford University Press.

Course Code	**ITMI0303**
Course Name	**Robot Process Automation (Elective)**
Contact Hours/ Credit Scheme	**L-T-P-C: 3-0-0-3**

Course Objectives:

1. To understand the basic concepts of Robotic Process Automation.
2. To expose to the key RPA design and development strategies and methodologies.
3. To learn the fundamental RPA logic and structure.
4. To explore the Exception Handling, Debugging and Logging operations in RPA.

Course Outcomes:

CO 1. Describe RPA, where it can be applied and how it's implemented.
CO 2. Describe the different types of variables, Control Flow and data manipulation techniques.
CO 3. Identify and understand Image, Text and Data Tables Automation.
CO 4. Describe how to handle the User Events and various types of Exceptions and strategies.
CO 5. Understand the Deployment of the Robot and to maintain the connection.

Course Outcomes	Program outcomes													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITMI0202										H				
CO 1	M	M			M								H	
CO 2	L			H										H
CO 3		H							L					
CO 4	H	M		H										
CO 5			H						H	H				

Course Content

Unit 1: Introduction to Robotic Process Automation: Emergence of Robotic Process Automation (RPA), Evolution of RPA, Differentiating RPA from Automation - Benefits of RPA - Application areas of RPA, Components of RPA, RPA Platforms. Robotic Process Automation Tools - Templates, User Interface, Domains in Activities, Workflow Files.

Unit 2: Automation Process Activities: Sequence, Flowchart & Control Flow: Sequencing the Workflow, Activities, Flowchart, Control Flow for Decision making. Data Manipulation: Variables, Collection, Arguments, Data Table, Clipboard management, File operations Controls: Finding the control, waiting for a control, Act on a control, UiExplorer, Handling Events

Unit 3: App Integration, Recording, Scraping, Selector, Workflow Activities. Recording mouse and keyboard actions to perform operation, Scraping data from website and writing to CSV. Process Mining.

Unit 4: Exception handling, Common exceptions, Logging- Debugging techniques, Collecting crash dumps, Error reporting. Code management and maintenance: Project organization, Nesting workflows, Reusability, Templates, Commenting techniques, State Machine.

TEXT BOOKS:

1. Learning Robotic Process Automation: Create Software robots and automate business processes with the leading RPA tool - UiPath by Alok Mani Tripathi, Packt Publishing, 2018.
2. Tom Taulli, “The Robotic Process Automation Handbook: A Guide to Implementing RPA Systems”, Apress publications, 2020.

REFERENCES:

1. Frank Casale (Author), Rebecca Dilla (Author), Heidi Jaynes (Author), Lauren Livingston (Author), Introduction to Robotic Process Automation: a Primer, Institute of Robotic Process Automation, Amazon Asia-Pacific Holdings Private Limited, 2018

2. Richard Murdoch, Robotic Process Automation: Guide To Building Software Robots, Automate Repetitive Tasks & Become An RPA Consultant, Amazon Asia-Pacific Holdings Private Limited, 2018
3. A Gerardus Blokdyk, “Robotic Process Automation Rpa A Complete Guide “, 2020

Course Code	**ITMI0305**
Course Name	**Data Science Programming (Elective)**
Contact Hours/ Credit Scheme	**L-T-P-C: 3-0-0-3**

Course Objectives

1. Learn about the basics of data Science and to understand the various supervised and unsupervised learning techniques.
2. Learn about the basic of data processing, cleaning, feature engineering
3. Bring together several key technologies used for manipulating, storing, and analyzing big data from advanced analytics perspectives.
4. Realize Hadoop architecture and implementation of MapReduce.

Course Outcomes

After the course completion, the student will be able to

CO 1. Clean/process and transform data.
CO 2. Analyze and interpret data using statistical approaches.
CO 3. Apply mathematical and statistical models to appropriately formulate and use data analyses.
CO 4. Formulate and use appropriate models of data analysis to solve real-world problem.

Course Outcomes	**Program outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITMI0305														
CO 1			M	H	M								H	M
CO 2	H	M	M	H	M								M	M
CO 3	H	H		H	M								M	M
CO 4	H	H	H		H	H						M	M	H

Course Content

Unit 1: Introduction to Data Science: The relationship between data science and Information Science, Facets of data, Tools for Data Science, Programming Basics (Python programming, R, Data Structures), Visualization/Plotting, Data Science Libraries (Pandas, PyPlot, matplotlib)

Unit 2: Descriptive Statistics: Introduction, Data Preparation, Exploratory Data Analysis, Estimation, Statistical Inference, Data Visualization, Data Wrangling, Feature Engineering.

Unit 3: Mathematical Foundations: Matrix Theory-Matrix Decomposition, Linear Algebra Fundamentals, Vector Spaces, Linear and Logistic Regression, Clustering, Probability, Hypothesis testing.

Unit 4: Advanced Databases: Introduction to databases, SQL, NOSQL, SQL vs NoSQL, NoSQL databases Monogodb, Neoj.

Unit 5: Advance topics: Network Analysis, Recommender Systems, Text Mining and Text Analysis, Introduction to Big Data, Handling Large Datasets, GPUs/CUDA programming, Parallel/distributed computing for data science (Map/Reduce, Spark/Hadoop), working on the cloud (Amazon Web services, Google Cloud Platform, Azure, etc).

Text/Reference Books

1. S. Chirag, A Hands-On Introduction To Data Science, Cambridge University Press, 2020.
2. S. Skiena, The Data Science Design Manual, Texts in Computer Science Springer, 2017.
3. J. Vander Plas, Python Data Science Handbook Essential Tools for Working with Data,
4. O'Reilly Media, Inc, Sebastopol, CA, 2016
5. J. Grus, Data Science from Scratch, O'Reilly, Beijing, 2015
6. L. Igual, and S. Seguí., Introduction to Data Science A Python Approach to Concepts, Techniques and Applications, Undergraduate Topics in Computer Science Springer, 2017

Sixth Semester

Course Code	**ITMI0302**
Course Name	**Software Engineering**
Contact Hours/ Credit Scheme	**L-T-P-C: 3-0-0-3**

Course Objectives

1. Study the current software engineering technique.
2. To understand software engineering methodologies for the development of Quality, cost-effective, schedule-meeting software.
3. Develop an understanding of ethical and professional issues related to Software engineering.

Course Outcomes

CO 1. Able to apply the concepts of software engineering.
CO 2. To analyze requirement techniques like Data flow diagram, Entity relationship diagram etc.
CO 3. Understanding the concept of Software Design.
CO 4. Demonstrate various testing methodologies and debugging tools for a prototype software.
CO 5. Develop various software reliability measures to assess the quality of software.

Course Outcomes	Program outcomes													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITMI0302														
CO1	M	M				L							M	
CO2		H			L					L			M	
CO3	H		H		M								H	
CO4			L	H	M				M					H
CO5	H	M	M		L									M

Course content

Unit 1: Introduction to Software Engineering: What is Software Engineering? Why Software Engineering, Software Crisis, Notable Changes in Software Development Practices, Software myths. The Software Process: Plan-driven and agile processes, different development philosophies: sequential vs iterative, software development life cycle (SDLC)

Unit 2: Agile-Software Development: Agility, Extreme Programming, Agile Project Management (Scrum), Scaling out and scaling up, Problem with Agile methods, Combining Agile and plan-driven methods, Agile methods across organizations Process models.

Unit 3: Software Requirements: Functional and non-functional requirements, User requirements, System requirements, Interface specification, the software requirements document.

Unit 4: Software Design Concepts: Activities carried out during design, Classification of Design Methodologies: Function-Oriented, Object-Oriented, Aspect-oriented, Component-based, Properties of a Good Design, Layered Design, Modularity, Function Oriented Design: Overview of SA/SD Methodology, Data Flow Diagrams (DFDs), DFD Model of a System, DFD Model to Structure Chart, Object-Oriented Design.

Unit 5: Testing Strategies: A strategic approach to software testing, test strategies for conventional software, Black-Box and White-Box testing, Validation testing, system testing, the art of Debugging. Product metrics: Software Quality, CASE Tools: Types of CASE tools, advantages and components of CASE tools, Unified Modelling Language (UML).

Recommended books:

5. Mall, R. (2021). Fundamentals of Software Engineering (6th ed.). PHI Learning.
6. Sommerville, I. (2022). Software Engineering (11th ed.). Pearson Education.
7. Jawadekar, W. S. (2021). Software Engineering: Principles and Practice (5th ed.). McGraw Hill Education.
8. Pressman, R. S., & Maxim, B. R. (2020). Software Engineering: A Practitioner's Approach (9th ed.). McGraw-Hill Education.

List of Electives for Seventh Semester

S.No.	Course Code	Courses	L	T	P	Credit
1.	ITMI0401	Java Programming Concepts	3	0	0	3
2.	ITMI0403	Introduction to Data Analytics	3	0	0	3
3.	ITMI0405	Prompt Engineering	3	0	0	3

Course Code:	**ITMI0401**
Subject Name	**Java Programming Concepts (Elective)**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**None**

Course Objectives

1. Building robust applications using Java's object-oriented features.
2. Understanding the usage of java class libraries.
3. Building multithreaded, platform-independent and GUI based java applications for business problems.

Course Outcomes:

CO 1. Write Java programs that solve simple business problems.
CO 2. Create java applications that are robust and multithreaded.
CO 3. Write simple GUI interfaces for a program to interact with users, and to understand the event-based GUI handling principles.

Course Outcomes	Program outcomes													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITMI0401														
CO 1	M	H	L				L						H	M
CO 2	H	L	M			M			H				M	M
CO 3	M	L	M				M						M	H
CO 4	H	M	H		L				L				M	H

Syllabus:

Unit 1: Introduction

Programming language Types and Paradigms, Computer Programming Hierarchy, How Computer Architecture Affects a Language ? Why Java ? ,Flavors of Java, Java Designing Goal, Role of Java Programmer in Industry, Features of Java Language, JVM –The heart of Java , Java's Magic Bytecode.

The Java Environment: Installing Java, Java Program Development , Java Source File Structure ,Compilation, Executions.

Basic Language Elements: Lexical Tokens, Identifiers, Keywords, Literals, Comments ,Primitive Datatypes, Operators Assignments.

Unit 2: Object Oriented Programming

Class Fundamentals , Object & Object reference , Object Life time & Garbage Collection, Creating and Operating Objects , Constructor & initialization code block, Access Control, Modifiers, methods Nested , Inner Class & Anonymous Classes, Abstract Class & Interfaces Defining Methods, Argument Passing Mechanism , Method Overloading, Recursion, Dealing with Static Members, Finalize() Method, Native Method. Use of "this " reference, Use of Modifiers with Classes & Methods, Design of Accessors and Mutator Methods Cloning Objects, shallow and deep cloning, Generic Class Types.

Unit 3: Extending Classes and Inheritance

Use and Benefits of Inheritance in OOP, Types of Inheritance in Java, Inheriting Data members and Methods , Role of Constructors in inheritance , Overriding Super Class Methods ,Use of "super" , Polymorphism in inheritance ,Type Compatibility and Conversion Implementing interfaces.

Package Organizing Classes and Interfaces in Packages , Package as Access Protection , Defining Package, CLASSPATH Setting for Packages, Making JAR Files for Library Packages Import and Static Import Naming Convention For Packages.

Unit 4: Exception Handling:

The Idea behind Exception ,Exceptions & 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. Array & String : Defining an Array, Initializing & Accessing Array, Multi –Dimensional Array, Operation on String, Mutable & Immutable String, Using Collection Bases Loop for String, Tokenizing a String, Creating Strings using StringBuffer .

REFERENCE BOOKS:

1. J.Nino and F.A. Hosch, "An Introduction to programming and OO design using Java", 2nd Ed., John Wiley & sons.
2. T. Budd, "An Introduction to OOP", 3rd Ed., Addison-Wesley
3. Y. Daniel Liang, "Introduction to Java programming", 6th Ed., Prentice Hall, 2006.
4. R.A. Johnson-Thomson , "An introduction to Java programming and object oriented application development", 1st Ed., Cengage Learning, 2006.
5. Cay.S.Horstmann and Gary Cornell, "Core Java 2, Vol 1, Fundamentals", 5th Ed., Prentice Hall, 2000.

Course Code:	**ITMI0403**
Subject Name	**Introduction to Data Analytics**
Contact hours/Credit Scheme (L-T-P-C)	**(L-T-P-C: 3-0-0-3)**
Pre-requisites	**None**

Course Objectives

1. Discuss the overall process of how data analytics is applied
2. Discuss the statistical approach for data analytics.
3. Discuss how data analytics can be used to better address and identify risks
4. Demonstrate the power of data analytics using case studies

Course Outcomes

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

CO 1. Understand applicability of data analytics for different application.
CO 2. Understand the necessity of randomness concept in practical situation
CO 3. Understand the importance of different hypothesis testing methods.
CO 4. Understand the importance of different statistical measures

Course Outcomes	Program outcomes													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITMI0403														
CO 1	M	H	L				L						H	M
CO 2	H	L	M			M			H				M	M
CO 3	M	L	M				L						M	H
CO 4	H	M	H		L				L				M	H

Course Content

Unit 1: Motivation for data analysis and interpretation: Examples from science, engineering, management and social sciences. Visualization of data. Different plots and graphs. Elementary statistical concepts related to data analysis, including Measures of Location and Measures of Variability.

Unit 2: Introduction to Probability Theory: Sample Spaces and events Probability axioms. Properties of Probability, Counting Techniques. Random Variables. Expectations and Variances. Conditional probabilities and conditional expectation. Independence. Important discrete and continuous distributions including the ones derived from the Normal (t, chi-squared and F). Random vectors. Joint Probability distributions. Bivariate and Multivariate Normal Distributions with the corresponding mean vectors, variance-covariance matrices and correlation matrices. Multinomial distribution. Distribution of the sample mean.

Unit 3: Estimation and Testing: Point and interval estimates of the unknown parameters. Unbiased estimators. Methods of point Estimation including maximum likelihood estimation. Confidence levels and confidence intervals. Hypothesis testing. Type 1 and Type 2 errors. Testing for parameters of a normal distribution and for percentages based on a single sample and based on two samples. Introduction to the chi-squared test. The concept of p-value. Mean-square estimation and Kalman filtering.

Unit 4: Regression and ANOVA: Simple linear regression. The least squares error criterion. Relation to maximum likelihood. The R^2 measure for fit of a regression model. Correlation coefficient. Regression models for prediction. Introduction to one way and two way ANOVA.

Text/Reference Books

1. S. M. Ross (2014) *Introduction to Probability and Statistics for Engineers and Scientists*, 5th Edition, Academic Press,
2. J. L. Devore (2015) *Probability and Statistics for Engineers*, 9th Edition, Brooks Cole.
3. D. Freedman, Robert Pisani, and Roger Purves (2007) *Statistics, 4th International Student Edition*, W.W. Norton & Company.
4. R.A. Irizarry (2019) *Introduction to Data Science, Data Analysis and Prediction Algorithms with R*, Chapman & Hall.
5. A.B. Downey (2014) *Think Stats* (*Exploratory Data Analysis in Python*), Green Tea Press.

Course Code:	**ITMI0405**
Subject Name	**Prompt Engineering**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**None**

Course Objective

This course will allow students to learn crafting, optimizing, and customizing prompts for various AI models.

Course Outcomes:

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

CO 1. Understand applicability of prompt engineering to real world.

CO 2. Understand the low code and no code changes.

CO 3. Understand the importance of citizen development.

CO 4. Understand the importance of low code and no code tools in real life scenario.

Course Outcomes	Program outcomes													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITMI0405														
CO 1	H										M		M	
CO 2	M			M							M	M		
CO 3	M	H		M		H	L	M			M	M	M	
CO 4			H	M	H	M	L	M	M	M	M	H	M	H

Course Content:

Unit 1: Workplace Technology: The struggle is real, Isolating cause and effect, Who's to blame for the status Quo?, The pandemic validated the importance of workplace Tech, Understanding the dual nature of contemporary IT, Retention issues and remote work.

Unit 2: Low-code/No-code changes everything: Building blocks and precursors, Distinguishing between No-Code and Low-Code tools, The major characteristics of Low-Code/ No-Code tools, The burgeoning industry, Major Low-Code/ No-Code subcategories, Contextualization Low-Code/ No-Code.

Unit 3: Introducing the citizen developer: History and definition, The rise of citizen developer, Common attributes of citizen developers, The benefits of citizen developer, IT benefits, Organizational benefits, Individuals and Team benefits.

Unit 4: Citizen development in Action: The municipality of rotterdam, A bubbly outcome for a budding entrepreneur, Low-Code/ No-Code powers up synergis education, Low-code/No-code transforms a family business,

Unit 5: How to evaluate Low-code/No-code tools and learn new one: Evaluating existing tools, Learning new Low-Code/ No-Code tools, Planning and gathering requirements, Design and development, Testing, Launch, Support, Maintenance, and Documentation.

REFERENCE BOOKS:

1. Phil Simon, Low-Code/No-Code: Citizen Developers and the Surprising Future of Business Applications, Racket Publishing, 2022

List of Electives for Eighth Semester

S.No.	Course Code	Courses	L	T	P	Credit

1.	ITMI0402	Business Intelligence	3	0	0	3
2.	ITMI0404	Wireless Communication for Beyond 5G Networks and IoT	3	0	0	3
3.	ITMI0406	Soft Computing Techniques	3	0	0	3

Course Code:	**ITMI0402**
Subject Name	**Business Intelligence**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objective:

1. To have a basic understanding of most recent advancements in Business Intelligence
2. To formulate the Business Intelligence models
3. To visualize techniques for effective application in Business intelligence.

Course Outcomes:

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

CO 1. To Understand the concept of Big Data and data analysis techniques.
CO 2. To learn the benefits for businesses and organizations.
CO 3. To design the storage of big data.
CO 4. To apply Big Data analytics to analyze and extract knowledge.

Course Outcomes	**Program outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITMI0402														
CO 1	H										M		M	
CO 2	M			M							M	M		
CO 3	M	H		M		H	L	M			M	M	M	
CO 4			H	M	H	M	L	M	M	M	M	H	M	H

Course Contents:

Unit 1: Introduction to Business Intelligence: Business View of IT Applications, Digital Data, OLTP vs. OLAP, Why, What and How BI? BI Framework and components, BI Project Life Cycle, Business Intelligence vs. Business Analytics.

Unit 2: Big Data Analytics: Big Data Analytics, Framework for Big Data Analysis, Approaches for Analysis of Big Data, ETL in Big Data, Introduction to Hadoop Ecosystem, HDFS, MapReduce Programming, Understanding Text Analytics and Big Data, Predictive analysis on Big Data, Role of Data analyst.

Unit 3: Business implementation of Big Data: Big Data Implementation, Big Data workflow, Operational Databases, Graph Databases in a Big Data Environment, Real-Time Data Streams and Complex Event Processing, Applying Big Data in a business scenario**.**
Unit 4: Security and Governance for Big Data, Big Data on Cloud, Best practices in Big Data implementation, Latest trends in Big Data, Latest trends in Big Data, Big Data Computation, More on Big Data Storage, Big Data Computational Limitations.

Recommended books:

1. Michael Minelli, Michele Chambers, Ambiga Dhiraj, Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses, Wiley CIO Series (2013),1st ed.

2. T. white, Hadoop: The Definitive Guide, O' Reilly Media (2012), 3rd ed.

Course Code:	**ITMI0404**
Subject Name	**Wireless Communication for Beyond 5G Networks and IoT**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**NONE**

Course Objectives

1. Able to understand different optical sources and wireless communications.
2. Recognize the feature and importance of different wireless communication and different modulation schemes.
3. Able to analyse the performance of different Optical wireless communication.

Course Outcomes

CO 1. Develop the understanding the Optical sources and Optical wireless communication systems
CO 2. Describe different modulation schemes.
CO 3. Describe the concept of different types of detectors, Indoor and Outdoor Optical wireless communication channel Modelling
CO 4. Analyse system performance for Indoor and Outdoor Optical wireless communication.

Course Outcomes	**Program outcomes**													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITMI0404														
CO 1	M				M				L		L			M
CO 2	L	L			M						L			
CO 3	L	L			L		L		L					L
CO 4	M	H	H		H	M					M			M

Course Content

Unit 1: Introduction: Optical Wireless Communication Systems; Existing wireless Access Schemes, OWC/Radio Comparison, Potential OWC Application Areas

Optical Sources: LEDs and Lasers (Internal and External Quantum Efficiency, Power and Luminous Efficiency, and Modulation Bandwidth)

Detectors: PIN and APD Photodetector, Photodetection Techniques, Photodetection Noise

Unit 2: Indoor Optical Wireless Communication Channel Modelling: LOS Propagation Model, Non-LOS Propagation Model, Interference from other other Light sources

Outdoor Optical Wireless Communication Channel Modelling: Atmospheric Channel Loss, Beam Divergence, Pointing Loss, Different Atmospheric Turbulence Models

Unit 3: Underwater Optical Wireless Communication Channel Modelling: Absorption, scattering, Turbulence, Multipath interference, Physical obstruction, and Background noise.

Modulation Schemes Digital Baseband Modulation Techniques like PAM, PPM, PIM etc., Multi-carrier Modulation (OFDM) for OWC, Color Shift Keying, NOMA etc.

Unit 4: System Performance Analysis: Indoor OWC links Effect of Ambient Light Sources on Indoor OWC Link Performance, Link Performance for Multipath Propagation

System Performance Analysis: Outdoor OWC links FSO Link Performance under the Effect of Atmospheric turbulence, Atmospheric Turbulence-Induced Penalty and mitigation strategies

Recommended Books

1."Advanced Optical Wireless Communication Systems" Shlomi Arnon, John Barry, George Karagiannidis, Robert Schober, and Murat Uysal... CAMBRIDGE UNIVERSITY PRESS
2.Optical Wireless Communications System and Channel Modelling Z. Ghassemlooy W. Popoola S. Rajbhandari... CRC Press
3.Optical Communications by Gerd Keiser

Course Code:	**ITMI0406**
Subject Name	**Soft Computing Techniques**
Contact hours/Credit Scheme (L-T-P-C)	**3-0-0-3**
Pre-requisites	**None**

Course Objectives

1. Motivation and historical background of Soft Computing.
2. Application of Fuzzy logic.
3. Biologically inspired algorithms such as neural networks, genetic algorithms, ant colony optimization, and bee colony optimization.
4. Hybrid systems of neural networks, genetic algorithms and fuzzy systems.

Course Outcomes

After the course completion, the student will be able to

CO1. Apply soft computing based solutions to real world and engineering problems.
CO2. Apply fuzzy logic and reasoning to handle uncertainty and solve engineering problems
CO3. Apply genetic algorithms to combinatorial optimization problems.
CO4. Apply neural networks to pattern classification and regression problems and compare solutions by various soft computing approaches for a given problem.

Course Outcomes	Program outcomes													
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
ITMI0406														
CO 1				M	H		L						H	H
CO 2		L		M			M	L					M	
CO 3						M		M					M	
CO 4	M					H			H				L	L

Course Content

Unit 1: Artificial Neural Networks: Basic concepts - Single layer perception - Multilayer Perception - Supervised and Unsupervised learning – Back propagation networks - Kohnen's self organizing networks - Hopfield network.
Unit 2: Fuzzy Systems: Fuzzy sets and Fuzzy reasoning - Fuzzy matrices - Fuzzy functions - Decomposition - Fuzzy automata and languages - Fuzzy control methods - Fuzzy decision making.
Unit 3: Neuro - Fuzzy Modeling: Adaptive networks based Fuzzy interface systems - Classification and Regression Trees - Data clustering algorithms - Rule based structure identification - Neuro-Fuzzy controls - Simulated annealing – Evolutionary computation.
Unit 4: Genetic Algorithms: Survival of the Fittest - Fitness Computations - Cross over - Mutation - Reproduction –Rank method - Rank space method
Unit 5: Soft computing and Conventional AI: AI search algorithm - Predicate calculus - Rules of interference – Semantic networks -Frames - Objects - Hybrid models - Applications.

Text/Reference Books

1. Jang J.S.R., Sun C.T. and Mizutani E, "Neuro-Fuzzy and Soft computing", Prentice Hall 1998
2. LaureneFausett, "Fundamentals of Neural Networks", Prentice Hall, 1994.
3. George J. Klir and Bo Yuan, "Fuzzy sets and Fuzzy Logic", Prentice Hall, USA 1995.
4. N. J. Nelsson, "Artificial Intelligence - A New Synthesis", Harcourt Asia Ltd., 1998
5. D.E. Goldberg, "Genetic Algorithms: Search, Optimization and Machine Learning", Addison Wesley, N.Y, 1989.