ADAMAS UNIVERSITY
SCHOOL OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
PG Program: MCA

COURSE STRUCTURE

Subject	Category	L-T-P-C	Credit	IA	TEE	Total Marks
Introduction to Programming	Major	3-0-0-3	3	30	70	100
Introduction to Programming-LAB	Major	0-0-4-2	2	50	50	100
Computer Organization & Architecture	Major	3-0-0-3	3	30	70	100
Computer Organization & Architecture-LAB	Major	0-0-4-2	2	50	50	100
Operating System	Major	3-0-0-3	3	30	70	100
Operating System-LAB	Major	0-0-4-2	2	50	50	100
Business Communication	HSSM-1	2-1-0-3	3	30	70	100
Numerical & Statistical Methods	Minor	3-0-0-3	3	30	70	100
Numerical & Statistical Methods-LAB	Minor	0-0-2-1	1	50	50	100
			22	**350**	**550**	**900**

Subject	Category	L-T-P-C	Credit	IA	TEE	Total Marks
Switching Circuits and Logic Design	Major	3-0-0-3	3	30	70	100
Object Oriented Programming with Java	Major	3-0-0-3	3	30	70	100
Object Oriented Programming with Java-LAB	Major	0-0-4-2	2	50	50	100
Data Structures	Major	3-0-0-3	3	30	70	100
Data Structures-LAB	Major	0-0-4-2	2	50	50	100
Database Management System	Major	3-0-0-3	3	30	70	100
Database Management System-LAB	Major	0-0-4-2	2	50	50	100
Discrete Mathematics	Minor	3-0-0-3	3	30	70	100

Python Programming-LAB	Minor	0-0-2-1	1	25	25	50
			22	**325**	**525**	**850**

Subject	**Category**	**L-T-P-C**	**Credit**	**IA**	**TEE**	**Total Marks**
Design and Analysis of Algorithms	Major	3-0-0-3	3	30	70	100
Data Communication & Computer Network	Major	3-0-0-3	3	30	70	100
Graph Theory	Major	3-0-0-3	3	30	70	100
Formal Language and Automata Theory Network Systems	Major	3-0-0-3	3	30	70	100
Discipline Specific Elective-1	Major	3-0-0-3	3	30	70	100
Discipline Specific Elective-1 -LAB	Major	0-0-4-2	2	50	50	100
Discipline Specific Elective-2	Major	3-0-0-3	3	30	70	100
Web Technology Lab	Minor	0-0-4-2	2	50	50	100
Mobile Applications using Android/IOS Lab	Minor	0-0-4-2	2	50	50	100
Project- 1	PW-minor	0-0-4-2	2	25	25	50
Total Credits:88			**26**	**355**	**595**	**950**

Subject	**Category**	**L-T-P-C**	**Credit**	**IA**	**TEE**	**Total Marks**
Discipline Specific Elective-3	Major	3-0-0-3	3	30	70	100
Compiler Design	Minor	3-0-0-3	3	30	70	100
Major Project- Internship Embedded Project work & Viva Voce (or) Research work, Dissertation & Thesis	PW-Major	0-0-24-12	12		200	200
			18	60	340	400

DSE-1 (To choose one course(T+P))	**Course Code**	**DSE-2 (To Choose one course)**	**Course Code**	**DSE-3 (To Choose one)**
Artificial Intelligence and Machine Learning		Natural Language Processing and Its Application		Public Block Chain

Fundamentals of Cloud Computing		Data Warehousing & Analytics		Cyber Security and Cryptography
Artificial Intelligence and Machine Learning-LAB				
Fundamentals of Cloud Computing-LAB				

CREDIT DISTRIBUTION (SEMESTER-WISE)

	Category	**S1**	**S2**	**S3**	**S4**	**Total**
Credit Distribution Summary	Major	15	18	20	3	56
	Minor	5	4	4	3	16
	HSSM-1	2				2
	Project-1-Minor			2		2
	Major project/OJT				12	12
	Total Credits					88

CREDIT DISTRIBUTION (YEAR-WISE)

YEAR I	**YEAR II**	**TOTAL**
44	44	88

MCA CREDIT DISTRIBUTION TABLE

Semester	**Lecture**	**Tutorial**	**Practical**	**Credits**
Semester 1	14	0	16	22

Semester 2	15	0	14	22
Semester 3	18	0	16	26
Semester 4	6	0	24	18
Total	53	0	70	88

1. DETAILED SYLLABI

	Introduction to Programming	**L**	**T**	**P**	**C**
Version 1.0	**Contact Hours – 45 Hours**	**3**	**0**	**0**	**3**
Pre-requisite/Exposure	**Knowledge on programming logic**				
Co-requisite	**NIL**				

Course Objectives:

- To motivate students to solve the problems in engineering using the concepts of procedural and object oriented programming.
- To enable students to apply OOP concepts in building solutions to real-world problems.
- To help the student to acquire knowledge of software development
- To enable students to debug simple C++ programs.
- To enable students to execute C++ programs successfully.

Course Outcomes:

On the completion of this course the student will be able to

- Discuss fundamentals of programming such as variables, conditional and iterative execution, methods, etc.
- Understand fundamentals of object-oriented programming in C++, including defining classes, invoking methods, using class libraries, etc.
- Explain important topics related to functions and pointers.
- Understand the scope of variables and utility of exception handling.
- Utilise the OOP knowledge to create, debug and run simple C++ programs.

Course Description:

This course introduces students to C and C++ programming language. Students will be taught the fundamentals of programming. These concepts are applicable to programming in any language. Topics covered include basic principles of programming using C++, algorithmic and procedural problem solving, program design and development, basic data types, control structures, functions, arrays, pointers, and introduction to classes for programmer-defined data types.

Course Content:

Unit-I	**09 Lecture Hours**
C Programming: Procedural programming, variables & data types, operators and conditional execution, understanding loops, arrays, types of arrays, functions, pointers, use of pointers with arrays, basic use of structures.	
Unit-II	**09 Lecture Hours**
Introduction to OOP: Need for OOP Paradigm, Procedural programming vs object oriented programming, object oriented concepts. **Class concept in OOP**: Difference between C structure and class, specifying a class, Defining member functions: inside and outside class, scope resolution operator, Array within a class, array of objects, Static data members and member functions, Object as function arguments, returning objects, Friend function, Constructor and destructor: Constructor, types of constructors: default, parameterized and copy constructor, constructor overloading, constructor with default parameter, dynamic initialisation of objects, destructor Operator overloading and Type Conversion: Defining operator overloading, overloading unary and binary operator, Data Conversion: Basic to User Defined, User defined to basic, Conversion from one user-defined to other.	
Unit-III	**09 Lecture Hours**
Functions: Main function, function prototyping, inline functions, reference variables, call by reference, Defaults arguments, function overloading, Math library functions. **Pointers**: memory allocation for objects, pointer to members, pointer to object, this pointer local classes.	
Unit-IV	**09 Lecture Hours**
Scope: Local and global scope, Inheritance and polymorphism: 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.** **Console IO operations**: C++ stream classes, Unformatted IO operations, formatted IO operations, managing output with manipulators. **Exceptions**: Run time errors, exception handling using try, catch and throw, working with files: Classes for file stream operations, opening and closing files,File opening modes, file Pointers, Error handling during file operations, command line arguments, templates	
Unit-V	**09 Lecture Hours**
Problem solving with C++: Case study for problem solving on various real life systems like Bank, Library, Hospital, Hotel, Employee management system etc.	

Text Books:

5. Bjrane Stroustrup, “C++ Programming language”, Pearson education Asia

Reference Books:

1. Yashwant Kenetkar,” Let us C++”, Oxford University Press
2. B.A. Forouzan and R.F. Gilberg, CompilerScience,” A structured approach using C++” Cengage Learning, New Delhi.

	Introduction to Programming Lab	L	T	P	C
Version 1.0	Contact Hours – 45 Hours	0	0	4	2
Pre-requisite/Exposue	Knowledge on programming basics				
Co-requisite	NIL				

Course Objectives:

- To motivate students to solve the problems in engineering using the concepts of procedural and object oriented programming.
- To enable students to apply OOP concepts in building solutions to real-world problems.
- To help the student to acquire knowledge of software development
- To enable students to debug simple C++ programs.
- To enable students to execute C++ programs successfully.

Course Outcomes:

On the completion of this course the student will be able to

- CO1: Define classes, objects, members of a class and the relationships among them needed for a finding the solution to specific problem.
- CO2: Apply fundamentals of object-oriented programming in C++, including defining classes, invoking methods, using class libraries, etc.
- CO3: Explain important topics related to functions and pointers.
- CO4: Understand the scope of variables and utility of exception handling.
- CO5: Utilise the OOP knowledge to create, debug and run simple C++ programs.

Course Description:

This course introduces students to C and C++ programming language. Students will be taught the fundamentals of programming. These concepts are applicable to programming in any language. Topics covered include basic principles of programming using C++, algorithmic and procedural problem solving, program design and development, basic data types, control structures, functions, arrays, pointers, and introduction to classes for programmer-defined data types.

Course Content:

Unit-I	09 Lecture Hours
Write a C program to find factorial of a number. Write a C program to find roots of a quadratic equation. Write a C program to find whether the number is Armstrong.	
Unit-II	**09 Lecture Hours**

Write a C++ program that demonstrate the basic class program to get department, name and salary of an employee. Write a C++ program that to calculate area of circle, square, rectangle and triangle using switch case statements Write a C++ program to that accepts number from user and displays all the factors of that number.	
Unit-III	**09 Lecture Hours**
Write a C++ Program to swap two numbers using pointers. Write a C++ Program to add two numbers using pointers. Write a C++ Program to find length of string using pointer.	
Unit-IV	**09 Lecture Hours**
Write a C++ Program to show multiple inheritance Write a C++ Program to show multilevel inheritance Write a C++ Program to fetch the content of an existing file and display its contents.	
Unit-V	**09 Lecture Hours**
Write a C++ Program to read the name and roll numbers of students from keyboard and write them into a file and then display it. Define a class "Time" that contains following data members and member functions. Data members: 1. Hours 2. Minutes 3. Seconds Member Functions: 1. To get time from user 2. To display time on the screen 3. To calculate sum of two time objects. Write a C++ program that can read values of Time for two objects T1 and T2, calculate sum and display sum using defined member functions Create class "Sales" having following data members and member functions: Data Members: 1. Name of Salesman 2. Sales of Salesman Member functions to calculate commission 1. Commission is Rs. 10 per thousand if sales are at least Rs. 25000 or more 2. Commission is Rs. 5 otherwise. Write a C++ program that calculate and print name and sales of salesman.	
Text Books: **45.** Bjrane Stroustrup, "C++ Programming language" , Pearson education Asia **Reference Books:** **3.** Yashwant Kenetkar," Let us C++, Oxford University Press **4.** B.A. Forouzan and R.F. Gilberg, Compiler Science," A structured approach using C++" Cengage Learning, New Delhi.	

	Computer Organization & Architecture	L	T	P	C
Version 1.0	Contact Hours – 45 Hours	3	0	0	3
Pre-requisite/Exposue	Basic computer Skills				
Co-requisite	--				

Course Content:

To study the basic organization and architecture of digital computers (CPU, memory, I/O, software). Discussions will include digital logic and microprogramming. Such knowledge leads to better understanding and utilization of digital computers, and can be used in the design and application of computer systems or as foundation for more advanced computer-related studies

Course Outcomes:

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

- CO1. Define functional block of a computer and relate data representation.
- CO2. Explain and understand memory hierarchy design, memory access time formula, performance improvement techniques, and trade-offs.
- CO3. Illustrate pipelined execution, parallel processing and principles of scalable performances.
- CO4. Analyse the concepts of memory utilization in a computer system.
- CO5. Define the implementation of parallel processors and analyse the synchronization techniques

Catalogue Description:

The architecture of computer systems and associated software. Topics include addressing modes, interrupt systems, input/output systems, external memory systems, assemblers, loaders, multi programming, performance evaluation, and data security.

Course Content:

Unit-I	09 Lecture Hours

Functional blocks of a computer: CPU, memory, input-output subsystems, control unit. Instruction set architecture of a CPU – registers, instruction execution cycle, RTL interpretation of instructions, addressing modes, instruction set. Case study – instruction sets of some common CPUs. **Data representation:** signed number representation, fixed and floating-point representations, character representation. Computer arithmetic – integer addition and subtraction, ripple carry adder, carry look-ahead adder, etc. multiplication – shift-and-add, Booth multiplier, carry save multiplier, etc. Division restoring and non-restoring techniques, floating point arithmetic

Unit-II	09 Lecture Hours

Operation in Peripheral devices and their characteristics:
Introduction to x86 architecture.

CPU control unit design: hardwired and micro-programmed design approaches, Case study – design of a simple hypothetical CPU.

Memory system design: semiconductor memory technologies, memory organization. **Peripheral devices and their characteristics:** Input-output subsystems, I/O device interface, I/O transfers – program controlled, interrupt driven and DMA, privileged and non-privileged instructions, software interrupts and exceptions. Programs and processes – role of interrupts in process state transitions, I/O device interfaces – SCII, USB

Unit-III	09 Lecture Hours

Inter-process pipelining:

Pipelining: Basic concepts of pipelining, throughput and speedup, pipeline hazards. .

Unit-IV	09 Lecture Hours

Memory and File organazation:

Memory organization: Memory interleaving, concept of hierarchical memory organization, cache memory, cache size vs. block size, mapping functions, replacement algorithms, write policies

Unit-V	09 Lecture Hours

Modern Case study in Parallel processor:
Parallel Processors: Introduction to parallel processors, parallel computer models, principles of scalable performances, multiprocessors and multicomputer, message passing mechanism, scalable & Multithreaded dataflow architecture, Concurrent access to memory and cache coherency and synchronization techniques, GPU Processors.
.

Text Books:

"Computer Organization and Design: The Hardware/Software Interface", 5th Edition by David A. Patterson and John L. Hennessy, Elsevier.
"Computer Organization and Embedded Systems", 6th Edition by Carl Hamacher, McGraw Hill Higher Education.

Reference Books:

"Computer Architecture and Organization", 3rd Edition by John P. Hayes, WCB/McGraw-Hill

"Computer Organization and Architecture: Designing for Performance", 10th Edition by William Stallings, Pearson Education.
"Computer System Design and Architecture", 2nd Edition by Vincent P. Heuring and Harry F. Jordan, Pearson Education

	Computer Organization & Architecture Lab.	L	T	P	C
Version 1.0	**Contact Hours – 30 Hours**	**0**	**0**	**4**	**2**
Pre-requisite/Exposure	**Fundamentals of Computer Architecture.**				
Co-requisite	**NIL**				

Course Objectives:

To study the basic organization and architecture of digital computers (CPU, memory, I/O, software). Discussions will include digital logic and microprogramming. Such knowledge leads to better understanding and utilization of digital computers, and can be used in the design and application of computer systems or as foundation for more advanced computer-related studies.

Course Outcomes:

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

- CO1. Write VHDL & Verilog programs.
- CO2. Design Logic circuit & ALU
- CO3. Analyze logic circuit
- CO4. Implement memory management schemes and page replacement schemes.
- CO5. Simulate file allocation and organization techniques.

Catalogue Description:

The architecture of computer systems and associated software. Topics include addressing modes, interrupt systems, input/output systems, external memory systems, assemblers, loaders, multiprogramming, performance evaluation, and data security.

Course Content:

Implementation based on basic Logic Gates (AND, OR, NOT, NAND, NOR, XOR, XNOR)

Implementation based on Half adder and Full adder (using data flow, behavioral, structural modelling)

Implementation based on Half subtractor and Full subtractor (using data flow, behavioral, structural modelling)

Implementation based on Full adder using two half adders and Full subtractor using two half subtractors
Implementation based on multiplexer, demultiplexer, Encoder and Decoder
Implementation based on D Flip Flop, SR Flip Flop, JK Flip Flop, T Flip Flop
Implementation based on 4 Bit Register (using Structural modelling)
Implementation based on 4 Bit Comparator (using Behavioral modelling)
Implementation based on 4 Bit ALU

	Operating System	**L**	**T**	**P**	**C**
Version 1.0	**Contact Hours – 45 Hours**	**3**	**0**	**0**	**3**
Pre-requisite/Exposure	**Data structures, Programming Languages, and Computer Architecture.**				
Co-requisite	**NIL**				

Course Objectives:

- To understand the students to study the basic principles and functionality of operating systems
- To provide the students to identify the concepts of CPU scheduling, concurrent processes, deadlock
- To allow the students to identify the significance of memory management and virtual memory.
- To enhance the skill of students to identify the disk scheduling, file systems, and device management.
- To understand the students to explain the performance trade-offs inherent in advance OS implementation.

Course Outcomes:

On the completion of this course the student will be able to

- CO1: Understand functionalities and features of Operating System
- CO2: Analyzing various scheduling algorithms and threading concepts to identify a suitable algorithm for a Given criteria.
- CO3: Assessing various solutions for critical Section problem. Applying deadlock avoidance principles and Check for the occurrence of deadlock.
- CO4: Explain different memory management techniques and its uses. Structuring an overview of file Systems and mass storage
- CO5: Understand the functionalities of modern operating system like Android, oxygen, Windows11 etc.

Course Description:

The course will begin with an overview of the structure of computer operating systems. The purpose of this course is to provide students basic knowledge of operating systems, difference between the kernel and user modes, concepts of application program interfaces, methods and implementations of interrupts. Students are introduced to the schedulers, policies, processes, threads, memory management, virtual memory, protection, access control, and authentication. Students learn system calls in different popular operating systems used in the industry. Particular emphasis will be given to three major OS subsystems: process management (processes, threads, CPU scheduling, synchronization, and deadlock), memory management (segmentation, paging, swapping), and file systems; and on modern operating system architecture.

Course Content:

Unit-I	**09 Lecture Hours**
Introduction to operating System: Introduction: Concept of Operating Systems, Operating Systems Objectives and Functions, Generations of Operating systems, Types of Operating Systems, OS Services, System Calls, Structure of an OS - Layered, Protection and Security, Case study on UNIX and WINDOWS Operating System.	
Unit-II	**09 Lecture Hours**
Introduction to Process and Process Scheduling : Process Management – Process concept- process scheduling, operations, Inter process communication. Multi Thread programming models. Process scheduling criteria and algorithms (FCFS, SJF, Priority, RR, Multilevel queue Scheduling), and their evaluation.	
Unit-III	**09 Lecture Hours**
Inter-process Communication and Deadlock : Process synchronization, the critical- section problem, Peterson's Solution, synchronization Hardware, semaphores, classic problems of synchronization, monitors, Producer Consumer problem, Readers & Writers Problem, Dining Philosopher Problem . Deadlocks: Definition, Necessary and sufficient conditions for Deadlock, Deadlock Prevention, and Deadlock Avoidance: Banker's algorithm, Deadlock detection and Recovery.	
Unit-IV	**09 Lecture Hours**

Memory and File Management :

Memory Management : Swapping, contiguous memory allocation, paging, structure of the page table, segmentation, Virtual memory, demand paging, page-Replacement, algorithms, Allocation of Frames, Thrashing.

File Management: Concept of File, Access methods, File types, File operation, Directory structure, File System structure, Allocation methods (contiguous, linked, indexed),Free-space management (bit vector, linked list, grouping), directory implementation (linear list, hash table), efficiency and performance.

Unit-V	**09 Lecture Hours**

Modern OS Architectures :
Case Study on: Android, Windows 11, Mac, oxygen OS and other contemporary Operating system.

Text Books:
Operating System Concepts- Abraham Silberchatz, Peter B. Galvin, Greg Gagne 9th Edition, John Wiley publishers, 2012

Operating Systems' – Internal and Design Principles, Stallings, Sixth Edition, Pearson education, 2005.

Reference Books:
Operating System a Design Approach-Crowley, 3 rd Edition, Tata Mcgraw Hill, 2009. Operating systems- A Concept based Approach-D.M.Dhamdhere, 2nd Edition, Tata Mcgraw Hill, 2012 Modern Operating Systems, Andrew S Tanenbaum 3rd edition Prentice-Hall, Inc, 2008

	Operating System Lab	**L**	**T**	**P**	**C**
Version 1.0	**Contact Hours – 45 Hours**	**0**	**0**	**4**	**2**
Pre-requisite/Exposure	**Data structures, Programming Languages, and Computer Architecture.**				
Co-requisite	**NIL**				

Course Objectives:

- To introduce basic Unix commands, system call interface for process management, interprocess communication and I/O in Unix.
- To understand the students to study the basic principles and functionality of operating systems.
- To provide the students to identify the concepts of CPU scheduling, concurrent processes, deadlock
- To allow the students to identify the significance of memory management and virtual memory.
- To enhance the skill of students to identify the disk scheduling, file systems, and device management.

Course Outcomes:

On the completion of this course the student will be able to

- **CO1: Understand** and implement basic services and functionalities of the operating system using system calls and shell script.
- CO2: Analyze and simulate CPU Scheduling Algorithms like FCFS, Round Robin, SJF, and Priority.
- CO3: Assessing various solutions for critical Section problem. Applying deadlock avoidance principles and Check for the occurrence of deadlock.
- CO4: Implement memory management schemes and page replacement schemes.
- CO5: Simulate file allocation and organization techniques.

Course Description:

The goal of this course is to have students understand and appreciate the principles in the design and implementation of operating systems software. The course will cover the concepts of operating systems, process management, memory management, file systems. Experiments on process scheduling and other operating system duties will be conducted through simulation/implementation in the laboratory.

Unit-I	09 Lecture Hours

Linux Commands/Shell Programming:
To study about the basics of Linux commands.

Implementation of shell scripting using conditional/branching statement.

Implementation of shell scripting using Loop statement.

Implementation of shell scripting using Array.

Implementation of shell scripting using String.

Implementation of shell scripting using Function and recursion.

Unit-II	09 Lecture Hours

Process Scheduling Algorithm:
Simulate the following non-preemptive CPU scheduling algorithms to find turnaround time and waiting time.

FCFS b) SJF c) Priority

Simulate the following non-preemptive CPU scheduling algorithms to find turnaround time and waiting time.

Shortest Remaining Time First b) Round Robin c) Priority

Simulate multi-level queue scheduling algorithm considering the following scenario. All the processes in the system are divided into two categories – system processes and user processes. System processes are to be given higher priority than user processes. Use FCFS scheduling for the processes in each queue.

Unit-III	09 Lecture Hours

Process Synchronization Problems /Deadlock:

Simulate producer-consumer problem using semaphores.

Simulate the concept of Dining-Philosophers problem.

Simulate Bankers algorithm for the purpose of deadlock avoidance.

Unit-IV	09 Lecture Hours

Memory Management Techniques:

Simulate page replacement algorithms

FIFO b) LRU c) Optimal

Simulate disk scheduling algorithms

FCFS b) SCAN c) C-SCAN

Simulate selection partition algorithm

a). Best Fit b). First Fit c). Worst Fit

Unit-V	09 Lecture Hours

File Organization Techniques:
simulate the following file organization techniques

a) Single level directory b) Two level directory c) Hierarchical

Text Books:

Operating System Concepts- Abraham Silberchatz, Peter B. Galvin, Greg Gagne 9th Edition, John Wiley publishers, 2012

Operating Systems' – Internal and Design Principles, Stallings, Sixth Edition, Pearson education, 2005.

	Business Communication	L	T	P	C
Version 1.2	Contact Hours – 45	2	1	0	3
Pre-requisites/Exposure	Understanding of significance of language				
Co-requisites					

Course Objectives

- To enable students to understand the basic principle of communication including the flow of communication, verbal as well as non- verbal in context of the organization.
- To enable describe the various ways of employment communication as well as develop the understanding and skill of presentation
- To provide to the students the basic understanding of the verbal and non - verbal communication so that they understand the different aspects of spoken and written business communication.
- To understand and apply basic principles of critical thinking, problem solving, and technical proficiency in the development of exposition and argument

Course Outcomes

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

- CO1: Explain the core principles of business communication, including verbal, non-verbal, and written formats, to develop effective communication strategies.
- CO2: Apply business communication techniques in professional settings, such as presentations, meetings, and negotiations, to convey information clearly and persuasively.
- CO3: Analyze various communication models and situations in business contexts, to identify barriers to effective communication and propose solutions.
- CO4: Design communication plans tailored to specific business scenarios, incorporating feedback and utilizing appropriate channels to achieve strategic goals.
- CO5: Evaluate the effectiveness of business communication practices by using feedback mechanisms and communication analytics, to enhance clarity, audience engagement, and organizational impact.

Course Description

Communication skill in a manager is one of the important skills, which a manager must possess to perform his/her role(s) effectively in an organization. Since he/she deals with employees, and with customers outside the organization, it is important that in an organization he should be well equipped in terms of different aspects of business communication. The course therefore covers all constituents, which will make a manager's job easy to handle.

Classroom activities involving lectures, discussions and case studies analysis (topped up with role-play) will be designed to encourage students to actually get involved, absorb and assimilate inputs. These activities will also be supplemented by group discussions, group presentations, cooperative group solving problems, analysis of video cases and debates. Class participation is a fundamental aspect of this course. Students will be encouraged to actively take part in all group activities and to give an oral group presentation. Students will be expected to interact with media resources, such as, web sites, videos, DVDs, and newspapers etc.

Course content UNIT
– I [5L]

Conceptual Issues in Communication, Principles of Communication, Process of Communication, Communication Networks in an Organization, Verbal and Non Verbal Communication.

UNIT II [**8L]** Barriers and Aids to Communication, The 7 C's and the 4 S's of Communication, Talk Tactics : Private and Public Speaking. Critical Reasoning : Theory and Caselets. The Framing of Arguments to Persuade ,Convince and Negotiate. Principles of Deductive and Inductive Principles in understanding Assumptions. Drawing Conclusions. Case Study Analysis in terms of Business and Current Affairs.

UNIT III [**8L]** Listening Process, Difference between Listening and Hearing.Deterrents to the Listening Process, The Positive Connotations of Good Listening, Case Study Analysis , Non Verbal Communication, Non Verbal Signifiers and Communication, Body Language and

Global Business Etiquette, Cross Cultural Communication, Case Study and Dramatic Practical on the above.

UNIT I V [8L] Written Communication, The 7 Cs of Written Communication, The First Draft and the Craft of Editing a write up. Memos, Letters, Emails, Net Etiquette and other Business Correspondence. Presentations: Preparing it and making an effective delivery. Practical Exercises.

UNIT V [8L] Negotiating Skills for Business. Telephone Culture and Video Conferencing. Group Discussion: Basics and Practice, Personal Interview: C.V. Format. Frequently Asked Questions and Mock Interviews.

	Numerical & Statistical Methods	L	T	P	C
Version 1.0	Contact Hour - 45	3	0	0	3
Pre-requisites/Exposure	Basic math Skills				
Co-requisites	--				

Course Objectives:

To provide students with knowledge in different computational errors occurred in numerical calculation. It can be minimized using different numerical techniques and, analyses and interpret statistical data using several statistical tools.

Course Outcomes:

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

- Find skewness, kurtosis, correlation coefficient and fit linear curve with the available set of data.
- Explain Baye's theorem for certainty of events and several probability distributions.
- Apply test of hypothesis to test mean, variance and different attributes for a population.
- Find real roots of algebraic and transcendental equations using Bisection method, Regula-Falsi method and Newton Raphson method.
- Utilize Euler method, Runge-Kutta method to obtain the solution to ordinary differential equations with initial conditions and, direct and iterative methods in simultaneous linear equation.
- Explain Numerical integration to obtain the value of an integral with finite limit and, finite differences to obtain interpolating and extrapolation values.

Catalogue Description:

This course introduces basic concepts in programming language to solve numerical and statistical problems. All the lectures will be devoted on discussions of basic theories and advanced topics, focusing on practical implementation of knowledge. Classes will be conducted by lecture as well as power point presentation, audio visual virtual lab session. The tutorials will familiarize the students with practical problem-solving techniques led by the course coordinator. Students will strongly grab the basic concepts of the subject via exercise and discussions with the coordinator.

Course Content:

Unit-I	09 Lecture Hours
Introduction to statistics:	

Statistics: Definition, scope and limitation, presentation of data, diagrammatic and graphical representation of data, measures of central tendency, mean, median and mode, geometric and harmonic mean and their limitations, Measure of variations, Range, Quartile, Variance, Standard deviation, Skegness, moment and Kurtosis.

Correlation and Regression:

Introduction to Correlation analysis, Karl Pearson correlation coefficient, Rank Correlation, Regression Analysis, Fitting Straight Lines, Method of least square, regression coefficients, properties of regression coefficients and applications

Unit-II	09 Lecture Hours

Introduction to Probability Distributions :

Probability: Introduction, Probability of an event, additive rule & multiplication rule, conditional probability Bayes' rule and applications.

Probability Distributions: Random variable, discrete and continuous probability distribution, Mathematical expectation, Variance of a random variable, Binomial, Hyper-geometric, Poisson distribution, Uniform, Normal, Exponential Distribution.

Test of hypothesis: Introduction, null hypothesis and alternative hypothesis, type I and type II errors, one and two tailed tests, test on a single mean when variance is known

.

Unit-III	09 Lecture Hours

Operation in Numerical solution of algebraic and transcendental equations :

Numerical solution of algebraic and transcendental equations: Introduction, Concept of Errors, Bisection Method, False Position Method, Secant Method, Newton-Raphson Method, Successive Approximation Method, Discussion of Convergence,

Solution of simultaneous linear equations: Gauss elimination method, pivoting, ill conditioned equations, Gauss Seidel and Gauss Jacobi iterative methods

Unit-IV	09 Lecture Hours

Finite difference analysis :

Finite difference analysis: Interpolation and Extrapolation, Calculus of difference, Newton's Forward Interpolation Formula and Backward Interpolation Formula, Lagrange's method, Newton's divided difference formula, Inverse Interpolation and its applications.

vector, linked list, grouping), directory implementation (linear list, hash table), efficiency and performance.

Unit-V	09 Lecture Hours

Case study in Numerical differentiation and integration:
Numerical differentiation and integration: Differentiation formulae based on polynomial fit, integration by trapezoidal and Simpson's one-third rules.Solution of simultaneous linear equations and ordinary differential equations: Euler methods,

	Numerical & Statistical Methods Lab	L	T	P	C
Version 1.0	**Contact Hour -45**	**0**	**0**	**2**	**1**
Pre-requisites/Exposure	**Basic math Skills**				
Co-requisites	**--**				

Course Objectives:

To provide students with computing knowledge in Numerical and Statistical problems using programming language.

Course Outcomes:

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

- CO1. Find real roots of algebraic and transcendental equations using Bisection method, Regula-Falsi method and Newton Raphson method.
- CO2. Solve system of linear equations using direct method and iteration method.
- CO3. Illustrate several methods of finite differences to obtain interpolating and extrapolating values from a set of data using.
- CO4. Classify Trapezoidal rule and Simpson's 1/3rd rule to obtain the value of an integral with finite limit.
- CO5. Utilize Euler method, Runge-Kutta to obtain the solution to ordinary differential equations with initial conditions.
- CO6. Find mean, variance, correlation coefficient and fit linear curve with the available set of data.

Course Description:

This course introduces basic concepts in programming language to solve numerical and statistical problems. All the lectures will be devoted on discussions of basic theories and advanced topics, focusing on practical implementation of knowledge. Classes will be conducted by lecture as well as power point presentation, audio visual virtual lab session. The tutorials will familiarize the students with practical problem-solving techniques led by the course coordinator. Students will strongly grab the basic concepts of the subject via exercise and discussions with the coordinator.

Course Content:

List of experiments Write and execute C-code for the following programs:

Sl No	Name of the experiment
1	To find real roots of algebraic and transcendental equations using Bisection method.
2	To find real roots of algebraic and transcendental equations using Regula-Falsi method.
3	To find real roots of algebraic and transcendental equations using Newton Raphson method.
4	To find solution of system of simultaneous algebraic equations using Gauss elimination method.
5	To find solution of system of simultaneous algebraic equations using Gauss-Seidal iterative method.
6	To find interpolating values using Newton's Forward interpolation formula.
7	To find interpolating values using Newton's Backward interpolation formula.
8	To find interpolating values using Lagrange's interpolation formula.
9	To evaluate integral value of a given function using Trapezoidal rule for numerical integration
10	To evaluate integral value of a given function using Simpson's $1/3^{rd}$ rule for numerical integration
11	To find numerical Solution of ordinary differential equation using Euler's method.
12	To find numerical Solution of ordinary differential equation using 4^{th} order Runge Kutta method.
13	To calculate mean and variance from a statistical data set.
14	To calculate Correlation coefficient.
15	To fit a linear curve using available data set.

	Switching Circuit and Logic Design	L	T	P	C
Version 1.0	Contact Hour -45				
Pre-requisites/Exposure	Basic Electronics, Modern Physics				
Co-requisites	Digital Electronics	3	0	0	3

Course Objectives:

- To introduce an overview of logic families.
- To develop students for building k-map.
- To provide the students a detailed analysis of sequential circuit.
- To introduce the students to formalize with ASM chart.

Course Outcomes:

On the completion of this course the student will be able to

- Understand and construct the basic design principles of logic gate.
- Understand the different fabrication techniques used in Bipolar, CMOS and PLA.
- Formalize with mealy and Moore machine.
- Construct ROM design.

Course Description:

The world of electronics is a lot easier to understand if we start by dividing it into two distinct categories: the “analog” world and the “digital” world. The analog world generally refers to any natural phenomenon that varies its own properties over a period of time. Take the outside temperature, for example. We notice that it changes rather slowly throughout the day, and at any instant we can measure how hot or cold it really is by using a simple thermometer.
The same changing properties can be observed, measured, and recorded in other natural phenomenon such as barometric pressure, wind speed, solar radiation, etc. If you were to record and graph each of the above events over a 24 hour period, you would notice one similar characteristic: the physical properties of each phenomenon change over time.

Course Content:

Unit-I	7 Lecture Hours
Switching Circuits: Logic families: TTL, nMOS, CMOS, dynamic CMOS and pass transistor logic (PTL) circuits, inverters and other logic gates, area, power and delay characteristics, concepts of fan-in, fan-out and noise margin.	
Unit-II	**10 Lecture Hours**
Switching theory: Switching algebra, logic gates, switching functions, truth tables and switching expressions, minimization of completely and incompletely specified switching functions, Karnaugh map and Quine-McCluskey method, multiple output minimization, representation and manipulation of functions using BDD's, two-level and multi-level logic circuit synthesis.	
Unit-III	**7 Lecture Hours**
Combinational logic circuits: Realization of Boolean functions using NAND/NOR gates, Decoders, multiplexers. logic design using ROMs, PLAs and FPGAs. Case studies, fault diagnosis of combinational circuits	
Unit-IV	**15 Lecture Hours**

Sequential circuits: Clocks, flip-flops, latches, counters and shift registers, finite-state machine model, Mealy and Moore machines, synthesis of synchronous sequential circuits, Conversion of Mealy m/c to Moore m/c and vice-versa, minimization and state assignment, Incompletely specified m/c's, asynchronous sequential circuit synthesis.	
Unit-V	**6 Lecture Hours**
ASM charts: Representation of sequential circuits using ASM charts, synthesis of output and next state functions, data path control path partition-based design.	
Text Books: H. Taub and D. Schilling, Digital Integrated Electronics, McGraw-Hill. **Reference Books:** Z. Kohavi, Switching and Finite Automata Theory, Tata McGraw-Hill Randy H. Katz and Gaetano Borriello, Contemporary Logic Design, Prentice Hall of India	

	Object Oriented Programming with Java	**L**	**T**	**P**	**C**
Version 1.0	**Contact Hours -45**	**3**	**0**	**0**	**3**
Pre-requisites/Exposure	**Basic concept of programming**				
Co-requisites	**--**				

Course Objectives:

Students will be motivated to solve the problems in engineering using the concepts of object-oriented programming.

Course Outcomes:

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

- CO1. Define Abstraction in all forms and in a holistic way
- CO2. Illustrate object oriented modelling techniques like classes and Instances modelling techniques
- CO3. Solve programs using standard design patterns
- CO4. Interpret fundamentals of object-oriented programming in Java, including defining Classes, invoking methods, using class libraries, etc.
- CO5. Construct programming solutions with exception handling and multi-threading concept CO6. Solve GUI program with proper event handling techniques

Catalog Description:

This course investigates object-oriented methods including object-oriented programming methodologies and techniques. Current methodology is emphasized. The use of object-oriented features such as encapsulation, information hiding, inheritance and polymorphism is reinforced by class assignments and programming exercises. The importance of multi-threading and exception handling is introduced in this course.

Course Content:

Unit I: 09 lecture hours

Introduction to oop concepts:

OOP Concepts - Data Abstraction, Encapsulation, Inheritance, Benefits of Inheritance, Polymorphism, Classes and Objects, Procedural and OOP Paradigms. Introduction To Java, Data Types, Variables & Constants, Scope & Life Time Of Variables, Precedence Of Operator, Expressions, Type Casting, Enumerated Types, Block Scope, Control Flow, Conditional Statements, Loops, Break & Continue Statements, Arrays, Console Input/Output, Formatting Output, Constructors Methods, Parameter Passing, Static Fields & Methods, Access Control, "This" Reference, Method Overloading, Recursion, Garbage Collection, Building Strings, String Class.

Unit II: 09 lecture hours

Idea on Inheritance and polymorphism Concepts:
Inheritance - Hierarchical Inheritance: Super And Sub Classes, Member Accessing Rules, Super Keyword, And Preventing Inheritance: Final Classes And Methods, Object Class And Its Methods.

Polymorphism - Dynamic Binding, Method Overriding, Abstract Classes and Methods
Interfaces - Interfaces and Abstract Classes, Definition, Implementation, Accessing Implementations by Interface References, Extending Interfaces.

Inner Classes - Usage, Local, Anonymous and Static Inner Classes, Examples.

Packages - Definition, Creation And Accessing A Package, Understanding CLASSPATH,

Importing Packages.

Unit III: 09 lecture hours

Brief operation in Exception Handling Concepts:

Exception Handling - Dealing With Errors, Advantages Of Exception Handling, The Classification - Exception Hierarchy, Checked And Unchecked Exceptions, Try, Catch, Throw, Throws And Finally, Exceptions-Throwing, Exception Specification, Built In Exceptions, Creating Exception Sub Classes.

Multithreading - Difference Between Multiple Processes And Multiple Threads, Thread States, Creating And Interrupting Threads, Thread Priorities, Synchronizing Threads, Inter-Thread Communication, Procedure Consumer Pattern.

Unit IV: 09 lecture hours

Brief operation in Connecting To Database:

Collection Framework - Introduction, Generics and Common Use Of Collection Classes,

Array List, Vector, Hash Table, Stack, Enumeration, Iterator, String Tokenizer, Random,

Scanner, Calendars And Properties.

Files - Streams - Byte Streams, Character Streams, Text Input/Output, Binary Input/Output, Random Access of File Operations, File Management.

Connecting To Database – JDBC / ODBC Type 1 To 4 Drivers, Connection And Handling Databases With JDBC.

Unit V: 09 lecture hours

Case study in GUI Programming:

GUI Programming - The AWT Class Hierarchy, Introduction To Swing, Swing Vs, AWT, Hierarchy Of Swing Components, Containers - Jframe, Japplet, Jdialog, Jpanel, Overview Of Swing Components: Jbutton, Jlabel, Jtextfield, Jtextarea, Swing Applications, Layout Management - Types - Border, Grid And Flow

Event Handling - Events, Sources, Classes, Listeners, Event Sources And Listeners, Delegation Event Model, Examples. Handling Mouse Events, Adapter Classes.

Applets - Inheritance Hierarchy For Applets, Differences Between Applets And Applications, Life Cycle, Passing Parameters To Applets, Applet Security Issues.

Text Books:

74. “Java Fundamentals - A Comprehensive Introduction”, Illustrated Edition By Daleskrien, Herbert Schildt, Mcgraw-Hill Education.

Reference Books:

1.“Java For Programmers”, 2nd Edition By Paul Deitel And Harvey Deitel, Pearson Education.
“Thinking In Java”, Low Price Edition By Bruce Eckel, Pearson Education

	Object Oriented Programming with Java Lab	**L**	**T**	**P**	**C**
Version 1.0	**Contact Hours -30**	**0**	**0**	**4**	**2**
Pre-requisites/Exposure	**Set Theory, Knowledge of programming language.**				
Co-requisites	--				

Course Objectives:

To understand how to design, implement, test, debug, and document programs that use basic data types and computation, simple I/O, conditional and control structures, string handling, functions and object oriented approaches.

Course Outcomes:

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

- CO1. **Define** classes, objects, members of a class and the relationships among them needed for a finding the solution to specific problem.
- CO2**. Illustrate** object oriented modelling techniques like classes and Instances modelling techniques
- CO3**. Solve** programs using standard design patterns.
- CO4**. Interpret** fundamentals of object-oriented programming in Java, including defining
- Classes, invoking methods, using class libraries, etc.
- CO5**. Construct** programming solutions with exception handling and multi-threading concept
- CO6**. Solve** GUI program with proper event handling techniques.

Catalogue Description:

This course investigates object-oriented methods including object-oriented programming methodologies and techniques. Current methodology is emphasized. The use of object-oriented features such as encapsulation, information hiding, inheritance and polymorphism is reinforced by class assignments and programming exercises. The importance of multi-threading and exception handling is introduced in this course.

Course Content:

List of Programs:

1. Assignments based on class, constructor.
2. Assignments based on overloading.
3. Assignments based on inheritance, overriding.
4. Assignments based on wrapper class, arrays.
5. Assignments based on developing interfaces- multiple inheritances, extending interfaces
6. Assignments based on creating and accessing packages
7. Assignments based on multithreaded programming
8. Assignments based on applet programming
9. Design a applet in Realtime web page design
10. Apply the concept in multi-threading
11. Apply the analysis in Java application development 108. Design a case study in android application.

Text Books:

1. "Java Fundamentals - A Comprehensive Introduction", Illustrated Edition By Daleskrien, Herbert Schildt, Mcgraw-Hill Education.

Reference Books:

1. "Java For Programmers", 2nd Edition By Paul Deitel And Harvey Deitel, PearsonEducation.
2. "Thinking In Java", Low Price Edition By Bruce Eckel, Pearson Education

		L	T	P	C
	Data Structures	**L**	**T**	**P**	**C**
Version 1.0	**Contact Hours – 45 Hours**	**3**	**0**	**0**	**3**
Pre-requisite/Exposure	--				
Co-requisite	--				

Course Objectives:

- To make students familiar with data structure
- To enable students to know and conceptualize stack, queue, linked list concept
- To enhance the skill of solving data problems over real time.

Course Outcomes:

On the completion of this course, the student will be able to

- **CO1: Define** the concept of Dynamic memory management, data types, and algorithms.
- **CO2: Illustrate** advantages and disadvantages of specific algorithms and data structures.
- **CO3: Solve** bugs in program, recognize needed basic operations with data structures.
- **CO4: Interpret** algorithms and data structures in terms of time and memory complexity of basic operations.
- **CO5: Compare** the computational efficiency of the principal algorithms for sorting, searching, and hashing.

Course Description:

Study of advanced programming topics focused on logical structures of data as well as the design, implementation and analysis of algorithms operating on these structures. Students will gain the fundamental concept of data structures and to emphasize the importance of data structures in developing and implementing efficient algorithms.

Course Content:

Unit-I: Introduction		**10 Lecture Hours**
Unit Heading: Concepts on Pointer, Dynamic Memory allocation, Structure, Recursion. Basic Terminologies: Elementary Data Organizations, Data Structure Operations: insertion, deletion, traversal etc.; Analysis of an Algorithm, Asymptotic Notations, Time-Space trade off. Arrays, 1D array, Operations on Array, 2D Array, Memory Representation of 2D Array, Operations on 2D Array, Searching, Linear Search, Binary Search		
Unit-II: Stacks and Queues		**8 Lecture Hours**
Unit Heading: ADT Stack and its operations, evaluation – Queue, Circular Queue, Priority Queue; Operations on ea	Applications of Stacks algorithms and . ADT queue, Types	: Expression Conversion and of Queue: Simple thms and their
Unit-III: Linked List		**10 Lecture Hours**
Unit Heading: Linked lists: Single, Doubly, Circular and Circular Doubly. Representation in memory, Algorithms of several operations : Traversing, Searching, Insertion into, Deletion from linked list (For all four kinds); Linked representation of Stack and Queue. Polynomial representation using Linked List.		
Unit-IV: Trees and Graphs		**12 Lecture Hours**
Unit Heading: Tree: Basic Tree Terminologies, Different types of Trees: Binary Tree, Threaded Binary Tree, Binary Search Tree, AVL Tree; Tree operations on each of the trees and their algorithms with complexity analysis. Applications of Binary Trees. Threaded Binary Tree. B Tree, B+ Tree: definitions, algorithms and analysis, Minimal Spanning Tree. **Graphs:** Basic Terminologies and Representations, Graph search and traversal algorithms and complexity analysis.		
Unit-V: Sorting and Hashing		**10 Lecture Hours**
Unit Heading: Objective and properties of different sorting algorithms: Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, Heap Sort; Performance and Comparison among all the methods, Hashing, Hash Functions, Collision, Collision Resolution Techniques.		
Text Books: Fundamentals of Data Structures, Illustrated Edition by Ellis Horowitz, SartajSahni and Computer Science Press.. Classic Data Structure, Debashis Samanta, PHI **Reference Books:** Data Structure Using C, Reema Thareja, Oxford University Press Algorithms, Data Structures, and Problem Solving with C++, Illustrated Edition by Mark Allen Weiss, Addison-Wesley Publishing Company.		

	Data Structures Lab	L	T	P	C
Version 1.0	Contact Hours – 45 Hours	0	0	4	2
Pre-requisite/Exposure	Concept in C-Programming				
Co-requisite	NIL				

Course Objectives:

- To teach programming (with an emphasis on problem solving)
- To introduce elementary data structures
- To allow student at a rudimentary level, be able to prove correctness (loop invariants, conditioning, etc).

Course Outcomes:

On the completion of this course the student will be able to

- CO1 : Explain asymptotic performance of the algorithms.
- CO2: Illustrate Linear data structures and their applications such as Stacks, Queues and Linked Lists
- CO3: Solve and understand Non-Linear Data Structures and their Applications such as Trees and Graphs
- CO4: Interpret searching and sorting algorithms

Course Description:

Data Structures (also called Data Structures and Algorithms in some places) is a core course in all computer science undergraduate curricula. The course is the basis for understanding several data structures and also algorithms that operate on them. The course forms the foundation for almost all computer science subjects: compilers, operating systems, databases, AI and software engineering.

Course Content:

Unit-I: Array	9 Lecture Hours
Write a program in C to Search an element into a 1D array. Write a program in C to sea into a 1D array using Binary Search.	

Write a program in C to insert an element into a 1D array. Write a program in C to delete an element from a 1D array. Write a program in C to add two given matrices. Write a program in C to multiply two given matrices. Write a program in C to find the transpose of a given matrix.	
Unit-II: Stacks and Queues	**8 Lecture Hours**
Write a program in C to implement Stack using Array . n Infix Expression to an g Stack. Write a program in C to implement a Qu Write a program in C to implement a Circular Queue .	
Unit-III: Linked List	**12 Lecture Hours**
Write a program in C to insert an element at the beginning into a Single Linked List Write a program in C to insert an element at the end into a Single Linked List Write a program in C to insert an element after a given element into a Single Linked List Write a program in C to insert an element at the beginning into a circular Linked List Write a program in C to insert an element at the end into a circular Linked List Write a program in C to insert an element at the beginning into a Doubly Linked List Write a program in C to insert an element at the end into a Doubly Linked List Write a program in C to insert an element after a given node into a Doubly Linked List Write a program in C to delete an element from a Single Linked List Write a program in C to delete an element from a Circular Linked List 11. Write a program in C to delete an element from a Doubly Linked List	
Unit-IV: Trees and Graphs	**11 Lecture Hours**
Write a program in C to implement a tree Write a program in C to implement a Binary Search Tree. Write a program in C to implement Minimal Spanning Tree Using Kruskal's Algorithm. Write a program in C to implement Minimal Spanning Tree using Prim's Algorithm. Write a program in C to implement Depth First Search 6. Write a program in C to implement Breadth First Search.	
Unit-V: Sorting	**5 Lecture Hours**
Write a program in C to implement Bubble Sort . C to implement Insertion Sor Write a program in C to im Write a program in C to im Write a program in C to implement Heap Sort.	

Text Books:
Chuck Lam, Hadoop in Action, Manning Publications,2010.
David Loshin, "Big Data Analytics: From Strategic Planning to Enterprise Integration with Tools, Techniques, NoSQL, and Graph", 2013.

Reference Books:
Big Data and Analytics by Seema Acharya, Subhashini Chellapan
Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses by Michael Minelli ,Michele Chambers , Ambiga Dhiraj

	Database Management System	**L**	**T**	**P**	**C**
Version 1.0	**Contact Hours -45**	**3**	**0**	**0**	**3**
Pre-requisites/Exposure	**Set Theory, Knowledge of programming language.**				
Co-requisites	--				

Course Objectives:

- To understand the different issues involved in the design and implementation of a database system.
- To study the physical and logical database designs, database modelling, relational, hierarchical, and network models.
- To understand and use data manipulation language to query, update, and manage a database
- To develop an understanding of essential DBMS concepts such as: database security, integrity, concurrency.
- To design and build a simple database system and demonstrate competence with the fundamental tasks involved with modelling, designing, and implementing a DBMS

Course Outcomes:

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

- CO1. Describe the fundamental elements of relational database management systems.
- CO2. Define the basic concepts of relational data model, entity-relationship model, relational database design, relational algebra and SQL.

- CO3. Design ER-models to represent simple database application scenarios.
- CO4. Build Structured Query Language (SQL) and apply to query a database.
- CO5. Improve the database design by normalization.
- CO6. Familiar with basic database storage structures and access techniques: file and page organizations, indexing methods including B tree, and hashing.
- CO7. Convert the ER-model to relational tables, populate relational database and formulate SQL queries on data.

Catalogue Description:

This is an introductory course in database management systems (DBMS) and file management systems. The course will cover the role of data, files and databases in information systems, data modeling concepts, data definition and manipulation using SQL, issues in data management and the development and implementation of database applications. Students will work in the Lab on various assignments including prototyping and SQL, utilizing state of the art DBMS and CASE tools.

Course Content:

Unit I: 8 lecture hours

Introduction to Database system architecture

Database system architecture: Data Abstraction, Data Independence, Data Definition Language (DDL), Data Manipulation Language (DML).

Data models: Entity-relationship model, network model, relational and object-oriented data models, integrity constraints, data manipulation operations.

ER models: Entity Set, Relation Ship Set, Cardinality Properties, Type of Entities, Type of Keys, Aggregation, Specialization and Generalization.

Unit II: 9 lecture hours

Introduction Relational query languages

Relational query languages: Relational algebra, Fundamental Operations, Additional Operations. Select, Project, Cartesian Product, UNION, Set difference, Rename. Types of joining operations, Division, Intersection, Aggregate. Tuple and domain relational calculus, SQL3, DDL and DML constructs, Open source and Commercial DBMS - MYSQL, ORACLE, DB2, SQL server.

Unit III: 10 lecture hours

Operation in , Dependency preservation, Lossless design

Relational database design: Integrity Constraint, Domain Constrain, Referential Integrity, Functional Dependencies, Closure of Set, Cover and Canonical Cover, Types of Anomalies, Armstrong's axioms, Extended Armstrong's axioms, Assertions and Demons.

Data Base Decomposition: Domain and data dependency, Normal forms: 1NF, 2 NF, 3 NF, BCNF, Dependency preservation, Lossless design.

Unit IV: 9 lecture hours

Analysis in Query processing and optimization

Query processing and optimization: Evaluation of relational algebra expressions, Query equivalence, Join strategies, Query optimization algorithms.

Storage strategies: Indices, B-trees, B+-trees, hashing, File System, Disk Organization, Physical Storage, Buffer management.

Unit V: 9 lecture hours

Case study in Database Security

Transaction processing: Failure, Recovery from Failure, Different States of Transaction, Transaction Isolation, ACID property, Serializability of scheduling, Multi-version and optimistic Concurrency Control schemes. Concurrency control: Locking and timestamp-based schedulers, 2-Phase Locking Protocol, Dead Lock, Database Security: Authentication, Authorization and access control, DAC, MAC and RBAC models, Intrusion detection, SQL injection. Advanced topics: Distributed databases, Data warehousing and data mining.

Text Books:

1. "Database System Concepts", 6th Edition by Abraham Silberschatz, Henry F. Korth, S. Sudarshan, McGraw-Hill
2. "Principles of Database and Knowledge – Base Systems", Vol 1 by J. D. Ullman, Computer Science Press.

Reference Books:

1. "Fundamentals of Database Systems", 5th Edition by R. Elmasri and S. Navathe, Pearson Education
2. "Foundations of Databases", Reprint by Serge Abiteboul, Richard Hull, Victor Vianu, AddisonWesley.

	Database Management System Lab	**L**	**T**	**P**	**C**
Version 1.0	**Contact Hours -45**	**0**	**0**	**4**	**2**
Pre-requisites/Exposure	**Set Theory, Knowledge of programming language.**				
Co-requisites	**--**				

Course Objectives:

- To understand the different issues involved in the design and implementation of a database system.
- To study the physical and logical database designs, database modelling, relational, hierarchical, and network models
- To understand and use data manipulation language to query, update, and manage a database
- To develop an understanding of essential DBMS concepts such as: database security, integrity, concurrency,
- To design and build a simple database system and demonstrate competence with the fundamental tasks

involved with modeling, designing, and implementing a DBMS.

Course Outcomes:

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

- CO1. Identify the use of Database Systems in different software and applications.
- CO2. Develop the queries using SQL in database creation interaction.
- CO3. Define a commercial relational database system (Oracle, MySQL) by writing SQL using the system.

Catalogue Description:

This course introduces the core principles and techniques required in the design and implementation of database systems. This introductory application-oriented course covers the relational database systems RDBMS - the predominant system for business scientific and engineering applications at present. It includes Entity-Relational model, Normalization, Relational model, Relational algebra, and data access queries as well as an introduction to SQL. It also covers essential DBMS concepts such as: Transaction Processing, Concurrency Control and Recovery. It also provides students with theoretical knowledge and practical skills in the use of databases and database management systems in information technology applications.

Course Content:

Experiment 1: Familiarization of structured query language.

Experiment 2: Table Creation.

Experiment 3: Insertion, Updation, Deletion of tuples.

Experiment 4: Executing different queries based on different functions.

Experiment 5: Performing joining operations.

Experiment 6: Nested Queries.

Experiment 7: Use of aggregate functions.

Experiment 8: Use of group functions.

Experiment 9: Use of order by functions.

Experiment 10: Arithmetic operations.

Experiment 11: Trigger using SQL.

Experiment 12: Introduction to PL/SQL.

Experiment 13: Report generation of various queries.

Experiment 14: Merging Data Bases with front end using ODBC connection.

Experiment 15: SQL Injection on a non-harmful test page.

Text Books:

1. "Database System Concepts", 6th Edition by Abraham Silberschatz, Henry F. Korth, S. Sudarshan, McGraw-Hill
2. "Principles of Database and Knowledge – Base Systems", Vol 1 by J. D. Ullman, Computer Science Press.

Reference Books:

1. "Fundamentals of Database Systems", 5th Edition by R. Elmasri and S. Navathe, Pearson Education
2. "Foundations of Databases", Reprint by Serge Abiteboul, Richard Hull, Victor Vianu, AddisonWesley.

	Discrete Mathematics	**L**	**T**	**P**	**C**
Version 1.0	**Contact Hours -45**	**3**	**0**	**0**	**3**
Pre-requisites/Exposure	**Set Theory, Knowledge of programming language.**				
Co-requisites	--				

Course Objectives:

- To develop an in-depth understanding of the algebraic structures like group, ring and field, combinatory, generating function, Recurrence relation, Graphs and Trees, mathematical logic.
- Students should be able to demonstrate application using the above mathematical tools in computer science related courses.

Course Outcomes:

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

- CO1. Define the fundamental knowledge to state the mathematical skills in Discrete Structure &Logic and allied fields.
- CO2. Define the fundamental knowledge to state the mathematical skills in basic and advance algebraic structures.
- CO3. Demonstrate basic concepts of combinatory including generating functions.
- CO4. Analyse the advance concept of graph theory in various mathematical fields.

Catalogue Description:

For any program related to Computer Science Discrete study of Mathematics is very much important. The purpose of this course is to understand and use (abstract) discrete structures and advance algebraic structure that are backbones of computer science. In particular, this course is meant to introduce logic, proofs, sets, relations, functions, counting, recurrence relation and graphs, with an emphasis on applications in computer science.

Course Content:

Unit I: [11 lecture hours]

Introduction to Sets, Relation and Function

Sets, Relation and Function: Operations and Laws of Sets, Cartesian Products, Binary Relation, Partial Ordering Relation, Equivalence Relation, Image of a function, Sum and Product of Functions, Injective, Surjective and Bijective functions, Composition of Functions, Inverse of functions.

Propositional Logic: Syntax, Semantics, Validity and Satisfiability, Basic Connectives and Truth Tables, Logical Equivalence: The Laws of Logic, Logical Implication, Rules of Inference.

Proof Techniques: Some Terminology, Proof Methods and Strategies, Forward Proof, Proof by Contradiction, Proof by Contraposition, Proof of Necessity and Sufficiency. Principles of Mathematical Induction: The Well-Ordering Principle, Recursive Definitions and Inductive proofs.

Unit II: [12 lecture hours]

Introduction to Advanced Counting Techniques

Size of a Set, Finite and infinite Sets, Countable and uncountable Sets, Cantor's diagonal argument and The Power Set theorem, Schroeder-Bernstein theorem.

The Division algorithm: Prime Numbers, The Greatest Common Divisor: Euclidean Algorithm, The

Fundamental Theorem of Arithmetic.

Advanced Counting Techniques: Recurrence relations and their solutions, Divide and Conquer Relations, Generating Basics of Counting, Pigeonhole Principle, Permutations and Combinations, Discrete Probability, Generalized Permutations and Combinations, Generating Permutations.
Functions, Inclusion-Exclusion Principle.

Unit III: [11 lecture hours]

Operation in Algebraic Structures and Morphism
Algebraic Structures and Morphism: Algebraic Structures with one Binary Operation, Semigroups, Monoids, Groups, Congruence Relation and Quotient Structures, Free and Cyclic Monoids and Groups, Permutation Groups, Substructures, Normal Subgroups, Algebraic Structures with two Binary Operation, Rings, Integral Domain and Fields, Boolean Algebra, Boolean Expression and Boolean Function, Identities of Boolean Algebra, Duality. Boolean Ring

Unit IV: [06 lecture hours]

Case study in Isomorphism, Eulerian and Hamiltonian Walks

Graphs and Trees: Graphs and their properties, Degree, Connectivity, Path, Cycle, Sub Graph, Isomorphism,

Eulerian and Hamiltonian Walks, Shortest Path Problems, Graph Colouring, Colouring maps and example

Unit V: [05 lecture hours]

Case study in planar graph

Planar Graphs, Colouring Vertices, Colouring Edges, List Colouring, Perfect Graph, Rooted trees, trees and sorting, weighted trees and prefix codes, Bi-connected component and Articulation Points, Spanning trees and Minimum Spanning Trees.

Text Books:

T1. Kenneth H. Rosen,Discrete Mathematics and its Applications, Tata McGraw - Hill.

T2. V Somasundaram, Discrete Mathematics with Graph Theory and Combinatory, Tata McGraw- Hill.

Reference Books:

R1. Norman L. Biggs, Discrete Mathematics, 2nd Edition, Oxford University Press.

R2. Discrete Mathematics for Computer Science", Illustrated Edition, Kenneth Bogart, Clifford Stein, Robert L. Drysdale, Key College Publishing.

	Design and Analysis of Algorithms	**L**	**T**	**P**	**C**
Version 1.0	**Contact Hours – 45 Hours**	**3**	**0**	**0**	**3**
Pre-requisite/Exposure	**Discrete Mathematics**				
Co-requisite	**Concepts on Programming, Logical Ability, Problem Solving**				

Course Objectives:

- To introduce problem solving approach through design.
- To develop students to analyse the existing algorithms and approach for improvement.
- To introduce the students a perspective to different design and analysis approach for algorithm(s) to solve a problem.
- To develop students to select optimal solution to a problem by choosing the most appropriate algorithmic method

Course Outcomes:

On the completion of this course the student will be able to

- Understand the basics about algorithms and learn how to analyse and design algorithms
- Choose brute force, divide and conquer, dynamic programming and greedy techniques methods to solve computing problems
- Understand the approach for solving problems using iterative method.
- Describe the solution of complex problems using backtracking, branch and bound techniques.
- Classify the different Computability classes of P, NP, NP-complete and NP-hard.

Course Description:

Algorithmic study is a core part of Computer Science. This study caters to all possible applicable areas of Computer Science. This study includes observation, design, analysis and conclusion. Various types of algorithms have different notion of implementation according to their cost (in terms their time and space complexity). This study also includes refinement of one algorithm as per the applicability to real problems. Categorization of algorithms according to different method of design also includes in this course. It also compares the same algorithm using different algorithm design methods. For example, Knapsack problem can be solved in Greedy approach and Dynamic approach, both are optimization method. This course enables the students to think analytically while applying, designing an algorithm to solve a specific problem.

Course Content:

Unit-I	**09 Lecture Hours**

Introduction: Characteristics of algorithm. Analysis of algorithm: Asymptotic analysis of complexity bounds – best, average and worst-case behaviour; Performance measurements of Algorithm, Time and space trade-offs, Analysis of recursive algorithms through recurrence relations: Substitution method, Recursion tree method and Masters' theorem. Algorithm Design Paradigms.

Unit-II	**09 Lecture Hours**
Sorting Algorithms & Data Structures: Selection sort, bubble sort, insertion sort, Sorting in linear time, count sort, Linear search, **Divide & Conquer:** Quick sort, worst and average case complexity, Merge sort, Matrix multiplication Binary search, Binary search tree, Strassen's algorithm for matrix multiplication, The substitution method for solving recurrences, The recursion-tree method for solving recurrences, The master method for solving recurrences.	
Unit-III	**09 Lecture Hours**

Greedy algorithms: General Characteristics of greedy algorithms, Problem solving using Greedy Algorithm- Activity selection problem, Minimum Spanning trees (Kruskal's algorithm, Prim's algorithm), Graphs: Shortest paths, The Knapsack Problem **Dynamic programming:** Introduction, The Principle of Optimality, Problem Solving using Dynamic Programming- Making Change Problem, Assembly Line Scheduling, Knapsack problem, Matrix chain multiplication, Longest Common Subsequence Dynamic Programming using Memoization.

Unit-IV	**09 Lecture Hours**

Graph Algorithms : Representations of graphs, Breadth-first search, Depth-first search, Topological sort, Strongly connected components, Minimum Spanning Trees, Growing a minimum-spanning tree, The algorithms of Kruskal and Prim, Single-Source Shortest Paths, Bellman-Ford algorithm, Single-source shortest paths in directed acyclic graphs, Dijkstra's algorithm, Difference constraints and shortest paths, Proofs of shortest-paths properties, All-Pairs Shortest Paths, Shortest paths and matrix multiplication, The FloydWarshall algorithm, Johnson's algorithm for sparse graphs, Maximum Flow, Flow-networks, The FordFulkerson method, **Branch & Bound & Backtracking**

Unit-V	**09 Lecture Hours**
String Matching The naive string-matching algorithm, The Rabin-Karp algorithm, String matching with finite automata, The Knuth-Morris-Pratt algorithm **Approximation Algorithms:** The vertex-cover problem, The traveling-salesman problem, The set-covering problem, Randomization and linear programming **NP-Completeness:** Polynomial time, Polynomial-time verification, NP-completeness and reducibility, NP-completeness proofs , NP-complete problems.	
Text Books: Introduction to Algorithms, 4TH Edition, Thomas H Cormen, Charles E Lieserson, Ronald L Rivest And Clifford Stein, MIT Press/ Mcgraw-Hill. Fundamentals of Algorithms – E. Horowitz Et Al. Algorithm Design, 1ST Edition, Jon Kleinberg and Évatardos, Pearson. Book 3 – Author – Publisher Reference Books: Algorithm Design: Foundations, Analysis, And Internet Examples, Second Edition, Michael T Goodrich And Roberto Tamassia, Wiley. Algorithms -- A Creative Approach, 3RD Edition, Udimanber, Addison-Wesley, Reading, MA.	

	Data Communication & Computer Network	**L**	**T**	**P**	**C**
Version 1.0	**Contact hour-45**	**3**	**0**	**0**	**3**
Pre-requisites/Exposure	**Computer Fundamentals**				
Co-requisites	--				

Course Objective:

- To become familiar with fundamentals of computer network.
- To become familiar with transmission media and data communication.
- To become familiar with addressing techniques and protocols.
- To become familiar with file transfer protocols, and concepts of secured data communication technique.

Course Outcomes:

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

- Explain key networking concepts, principles, design issues and techniques at all protocol layers.
- Contrast between different types of networks (e.g., wide area networks vs. local area networks, wired vs. wireless) in terms of their characteristics and protocols used.
- Describe different types of networked applications and what underlying network protocols are needed to meet their diverse requirements.
- Distinguish between control and data planes in computer networks, and their corresponding architectures in real-world networks (including the Internet).
- Illustrate reliable transport protocols and networked system architectures via implementation using Socket APIs, measurement and analysis.

Catalogue Description:

In this course, students will study architectures, protocols, and layers in computer networks and develop client-server applications. Topics include the OSI and TCP/IP models, transmission fundamentals, flow and error control, switching and routing, network and transport layer protocols, local and wide-area networks, wireless networks, client-server models, and network security. Students will extend course topics via programming assignments, library assignments and other requirements.

Course Content:

Unit I: 5 lecture hours

What Is the Internet?, Network Edge, Network Core, Delay, Loss, and Throughput in Packet-Switched Networks, Protocol Layers and Their Service Models, Networks Under Attack.

Unit II: 8 lecture hours

Principles of Network Applications, Web and HTTP, Electronic mail in Internet, DNS—The Internet's Directory Service, Peer-to-Peer Applications.

Unit III: 9 lecture hours

Introduction and Transport-Layer Services, Multiplexing and De-multiplexing, Connectionless Transport: UDP, Connection-Oriented Transport: TCP, Principles of Congestion Control, TCP Congestion Control

Unit IV: 9 lecture hours

Introduction, Virtual Circuit and Datagram Networks, Internet Protocol (IP): Forwarding and Addressing in the Internet, Routing Algorithms, Routing in the Internet, Routing in the Internet, Broadcast and Multicast Routing

Unit V: 9 lecture hours

Introduction to the Link Layer, Error-Detection and -Correction Techniques, Multiple Access Links and Protocols, Switched Local Area Networks, Link Virtualization.

Unit VI: 3 lecture hours

What Is Network Security? Principles of Cryptography

Unit VII: 2 lecture hours

What Is Network Management? Internet-Standard Management Framework

Text Books:

1. Computer Networking -Top Down Approach- James F. Kurose and Keith W. Ross-- Pearson 2013, sixth Edition
2. Data Communications and Networking- Behrouz A. Forouzan-McGraw-Hill 2007, fourth Edition.

Reference Books:

Data Networks- DimitriBertsekas and Robert Gallager- Prentice Hall, 1992
Computer Networks (5th Edition) – Andrew S. Tanenbaum, Pearson 2011

	Graph Theory	**L**	**T**	**P**	**C**
Version 1.0	**Contact Hours – 45 Hours**	**3**	**0**	**0**	**3**
Pre-requisite/Exposure	**Data Structures**				
Co-requisite	**--**				

Course Objectives:

- To understand and apply the fundamental concepts in graph theory.
- To apply graph theory-based tools in solving practical problems.
- To improve the proof writing skills.
- To state the theorems and prove formally using various techniques.
- To understand various graphs algorithms and analyse them.

Course Outcomes:

On the completion of this course the student will be able to

- CO1: Understand the different distance measures in graphs. Define the special types of graphs – complete graph, regular graph, bipartite graph and their properties.
- CO2: Discuss the properties of trees, Arboricity, vertex and edge connectivity,

auto-morphism groups, reconstruction problem and Mengers theorem.

- CO3: Distinguish between Eulerian and Hamiltonian Graphs. Demonstrate Euler's theorem, Kuratowski's theorem, Colouring of planar graphs, Crossing number and thickness.
- CO4: Explain Query processing and optimization. Analyze the storage strategies
- CO5: Differentiate among the matching factors, decomposition and domination in graph theory.

Course Description:

This course is aimed to cover a variety of different problems in Graph Theory with an emphasis on applications and modelling. Graph theory is a study of graphs, trees and networks. In this course students will come across a number of theorems and proofs. Theorems will be stated and proved formally using various techniques. Topics that will be discussed include Euler formula, Hamilton paths, planar graphs and coloring problem; the use of trees in sorting and prefix codes; useful algorithms on networks such as shortest path algorithm, minimal spanning tree algorithm and min-flow max-cut algorithm.

Course Content:

Unit-I	**8 Lecture Hours**
Basics: Graph – definition; Degree sequences, Different distance measures in graphs, Special types of graphs – complete graph, regular graph, bipartite graph and their properties.	
Unit-II Unit-II Structure and Symmetry: Cut vertices, bridges and blocks, auto-morphism groups, reconstruction problem. Trees and Connectivity: Properties of trees, Arboricity, vertex and edge connectivity, Mengers theorem.	**9 Lecture Hours**

Unit-III	**10 Lecture Hours**
Unit-III Eulerian and Hamiltonian Graphs: Characterization of Eulerian graphs, Sufficient Conditions for Hamiltonian graphs. Colouring and Planar Graphs: Vertex and edge colouring, perfect graphs, planar graphs, Euler’s theorem, Kuratowski’s theorem, Colouring of planar graphs, Crossing number and thickness.	
Unit-IV	**9 Lecture Hours**
Query processing and optimization: Evaluation of relational algebra expressions, Query equivalence, Join strategies, Query optimization algorithms. Storage strategies: Indices, B-trees, B+-trees, hashing,	

File System, Disk Organization , Physical Storage, Buffer management.	
Unit-V	**9 Lecture Hours**
Vert Matching, factors, decomposition and domination. External Graph Theory: Turan's theorem , Ramsay's theorem, Szemeredi's regularity lemma and their applications.	
Text Books: 1. Graph Theory, J. A. Bondy and U. S. R. Murthy, Sringer Verlag, 2008. 2. Introduction to Graph Theory, D. B. West, PHI, 2004.	
Reference Books: 1. Graph Theory, R. Diestel, Sringer Verlag, 2003.	

	Formal Language and Automata Theory	**L**	**T**	**P**	**C**
Version 1.0	**Contact Hours – 45 Hours**	**3**	**0**	**0**	**3**
Pre-requisite/Exposure	**NIL**				
Co-requisite	**NIL**				

Course Objectives:

- Introduce concepts in automata theory and theory of computation
- Identify different formal language classes and their relationships
- Design grammars and recognizers for different formal languages
- Prove or disprove theorems in automata theory using its properties
- Determine the decidability and intractability of computational problems

Course Outcomes:

On the completion of this course the student will be able to

- Define the basic concepts in formal language theory, grammars, automata theory, computability theory, and complexity theory.
- Demonstrate abstract models of computing, including deterministic (DFA), non-deterministic (NFA), Push Down Automata (PDA) and Turing (TM) machine models and their power to recognize the languages.
- Prove and disprove theorems establishing key properties of formal languages and automata.
- Acquire a fundamental understanding of core concepts relating to the theory of computation and computational models including (but not limited to) decidability and Intractability.
- Solve fundamental problems related to Computational Model.

Course Description:

This course will provide a foundation to the "Theory of Computation". The student will realize that the sometimes chaotic technology oriented world of computers has a very elegant mathematical basis to it. This basis is deeply rooted in mathematics developed before the days of modern computers. Our study will lead to some interesting implications concerning the theoretical limits of computing. On the practical side, this course is a background for a course on compilers. Topics covered in this course include: mathematical prerequisites, finite state machines (automata), concept of a language and grammars, deterministic and non-deterministic accepters, regular expressions and languages, contextfree languages, normal/canonical forms, pushdown automata, Turing machines, context sensitive languages, recursive and recursively enumerable languages. Each of the language classes has two points of view: a class of automata defining the language, and a class of grammars defining the language. This dual approach to defining languages, will finally lead to the Chomsky hierarchy of languages. We shall observe that the Turing Machine not only serves to define a language class, but also a mathematical model for computation itself and defines the theoretical limits of computation.

Course Content:

Unit-I	**09 Lecture Hours**

Mathematical Preliminaries:
Set Theory, Describing a Set, Empty Set, Identity and Cardinality, Subset, Power Sets, Operations on Sets: Union, Intersection, Set Theoretic Equalities, Sequence versus Set, Ordered Pairs, Cartesian Product,Relations, Binary Relation, Domain and Range of Relation, Operations on Relations, Properties of Relations, Functions, Types of Functions, Alphabet, String and Language, Operations on Language, Grammars, Types of Grammars–Chomsky Hierarchy, Graphs and Trees, Directed Graph, Undirected Graph, Trees, Lemma, Theorem Proving, Proof by Induction Proof by Contradiction, Proof by Example.

Unit-II	**09 Lecture Hours**

Finite Automata: Finite-state Machine, Finite-Automaton Model, Properties of Transition Function ‘c’, Transition Diagram, Transition Table, Language Acceptance, Two Types of Finite Automata, Deterministic Finite Automata (DFA) Non-deterministic Finite Automaton, Acceptance of NFA, Equivalence of DFAs and NFAs, Converting NFA to DFA, Subset Construction, NFA with Epsilon-(ε) Transitions, Epsilon Closure (e-closure), Eliminating e-Transitions, Converting NFA with ε-Transition to NFA, without εTransition, Converting NFA with ε-Transition to DFA, Comparison Method for Testing, Equivalence of Two FA, Reduction of Number of States in FA, Indistinguishable States, Equivalent Classes, Minimization of DFA, Minimization of DFA Using Myhill Nerode Theorem, Finite Automata with Output, Moore Machine, Mealy Machine, Equivalence Between Moore and Mealy Machines, Interconversions Between Machines, Applications of Finite Automata with Output, The Full-adder, The String Sequence Detector. **Regular Languages and Regular Grammar:** Regular language, Regular expressions, Deterministic finite automata (DFA) and equivalence with regular expressions, NFA and equivalence with DFA, Regular grammars and equivalence with finite automata, Properties of regular languages, Pumping lemma for regular languages, Problem solving using pumping lemma.	
Unit-III Pushdown Automata & Context Free Languages: Graphical Representation of PDA, Instantaneous Description of PDA, Language Acceptance by PDA, Equivalence of Acceptance of Final State and Empty Stack, Types of PDAs, Deterministic PDA, Closure Properties of DCFL, Decision Properties of DCFLs, DPDA and Regular Languages, DPDA and Ambiguous Grammar, Equivalence of PDA’s and CFG’s, Nondeterministic pushdown automata (NPDA), NPDA and equivalence with CFG, Constructing PDA for Given CFG, Constructing CFG for the Given PDA, Two-stack PDA, Applications of PDA, PDA as a Parser, Top-down Parser Using the PDA, Pumping lemma for context-free languages. Context Free Grammar: Context-free grammars (CFG), Leftmost and Rightmost Derivations, Derivation Tree, Equivalence of Parse Trees and Derivations, Ambiguous Grammar, Removing Ambiguity, Inherent Ambiguity, Simplification of Grammars, Elimination of Useless Symbols, Elimination of e-Productions, Eliminating Unit Productions, Chomsky normal forms, Greibach normal forms	**09 Lecture Hours**
Unit-IV Context Sensitive Grammar and Languages: Context-sensitive grammars (CSG), Context-sensitive Languages, Linear bounded automata, Linear bounded automata and equivalence with CSG, Properties of Context-sensitive grammars (CSG) and Languages, Properties of Linear bounded automata. Turing Machine: Turing Assumptions, Instantaneous Description, Turing Machine as Language Accepter, Turing Machine as a Computational Machine, Techniques for Turing Machine Construction, Storage in Finite Control, Multi-track Tape, Checking off Symbols, Subroutines, Shifting Over, Types of Turing Machines, Nondeterministic Turing Machines, Turing Machines with Two-dimensional, Tapes, Turing Machines with Multiple Tapes, Turing Machines with Multiple Heads, Turing Machines with Infinite Tape, Church’s Thesis, Turing Machines as Enumerators, Universal Turing Machine, Counter Machine, Recursive and Recursively Enumerable Languages, Unrestricted grammars and equivalence with Turing machines, Church-Turing thesis, universal Turing machine, Rice’s theorem, undecidable problems about languages	**09 Lecture Hours**
Unit-V Decidability: Decidable Languages, Decidable problems concerning regular languages, Decidable problems concerning context-free languages, Undecidability, The	**09 Lecture Hours**

diagonalization method, An undecidable language, A Turing-unrecognizable language. Reducibility: Undecidable Problems from Language Theory, Reductions via computation histories, Simple Undecidable Problem, Mapping Reducibility, Computable functions, Formal definition of mapping reducibility. Time Complexity: Measuring Complexity, Big-O and smallo notation, Analyzing algorithms, Complexity relationships among models, The Class P, Polynomial time, Examples of problems in P, The Class NP, Examples of problems in NP, P versus NP, NP-completeness, Polynomial time reducibility, Definition of NP-completeness, The Cook–Levin Theorem. Space Complexity: Savitch's Theorem, The Class PSPACE, PSPACEcompleteness	
Decidability: Decidable Languages, Decidable problems concerning regular languages, Decidable problems concerning context-free languages, Undecidability, The diagonalization method, An undecidable language A Turing-unrecognizable language **Reducibility:** Undecidable Problems from Language Theory, Reductions via computation histories, Simple Undecidable Problem, Mapping Reducibility , Computable functions, Formal definition of mapping reducibility **Time Complexity:** Measuring Complexity, Big-O and small-o notation, Analyzing algorithms, Complexity relationships among models, The Class P, Polynomial time, Examples of problems in P, The Class NP, Examples of problems in NP, P versus NP, NP-completeness, Polynomial time reducibility, Definition of NP-completeness, The Cook–Levin Theorem **Space Complexity:** Savitch's Theorem, The Class PSPACE, PSPACE-completeness	
Text Books: Introduction to Automata Theory, Languages, and Computation, 3rd Edition, John E. Hopcroft Rajeev Motwani and Jeffrey D. Ullman, Pearson Education. Michael Sipser, Introduction to the Theory of Computation, PWS Publishing An Introduction To Formal Languages And Automata, Peter Linz Reference Books: Harry R. Lewis and Christos H. Papadimitriou, Elements of the Theory of Computation,Pearson Education Asia. Dexter C. Kozen, Automata and Computability, Undergraduate Texts in Computer Science, Springer.	

	Compiler Design	**L**	**T**	**P**	**C**
Version 1.0	**Contact Hours – 45 Hours**	**3**	**0**	**0**	**3**
Pre-requisite/Exposure	**Finite Automata, Data structures, Computer Organization.**				
Co-requisite	**NIL**				

Course Objectives:

- To understand students how the process of language translation process.
- To interpret students the theory and practice of compiler implementation.
- To enhance the skills of student to implement lexical analysis, a variety of parsing techniques and semantic analysis of a programming language, along with error detection and recovery.
- To allow the students to identify the various storage allocation, code optimization techniques and code generation.
- To understand students the use of object code generation process

Course Outcomes:

On the completion of this course the student will be able to

- CO1: Understand the major phases of compilation, particularly lexical analysis.
- CO2: Understand the basic concepts of parsing and Design parser for a given languageusing top down and Bottom-up parser.
- CO3: Demonstrate the use of formal attributed grammars for specifying the syntax and semantics of Programming languages.
- CO4: Apply various optimization techniques for the design of a compiler.
- CO5: Understand the concepts of symbol tables and implement code generation techniques.

Course Description:

This course will teach students the fundamental concepts and techniques used for building a simple compiler. It will also discuss the major ideas used today in the implementation of programming language compilers, including lexical analysis, parsing, syntax-directed translation, abstract syntax trees, types and type checking, intermediate languages, dataflow analysis, program optimization, code generation, and runtime systems. As a result, you will learn how a program written in a high-level language designed for humans is systematically translated into a program written in low-level assembly more suited to machines. Along the way we will also touch on how programming languages are designed, programming language semantics, and why there are so many different kinds of programming languages.

Course Content:

Unit-I	09 Lecture Hours
Unit-I Introduction of Compilers and Lexical Analysis and Lex/Flex: Overview of language processing – pre-processors – compiler – assembler – interpreters, pre-processors, – linkers & loaders - structure of a compiler – phases of a compiler. Lexical Analysis – Role of Lexical Analysis – Lexical Analysis Vs Parsing – Token, patterns and Lexemes – Lexical Errors – Regular Expressions – Regular definitions for the language constructs – Strings, Sequences, Comments – Transition diagram for recognition of tokens, Reserved words and identifiers,	
Examples.	
Unit-II	**09 Lecture Hours**
Syntax Analysis and Yacc/Bison: Syntax Analysis – discussion on CFG, LMD,RMD, parse trees, Role of a parser – classification of parsing techniques – Brute force approach, left recursion, left factoring, Top down parsing – First and Follow- LL(1) Grammars, Non-Recursive predictive parsing – Error recovery in predictive parsing. Bottom up parsing, Types of Bottom up approaches. Introduction to simple LR – Why LR Parsers – Model of an LR Parsers – Operator Precedence- Shift Reduce Parsing – Difference between LR and LL Parsers, Construction of SLR Tables. More powerful LR parses, construction of CLR (1), LALR Parsing tables, Dangling ELSE Ambiguity, Error recovery in LR Parsing, Comparison of all bottoms up approaches with all top down approaches.	
Unit-III	**09 Lecture Hours**
Intermediate Code Generation: Semantic analysis, SDT Schemes, evaluation of semantic rules. Intermediate Code Generation: Intermediate languages, three address code, quadruples, triples, abstract syntax trees. Types and declarations, Assignment statements, Boolean expressions, Case statements, Back Patching, Procedure calls type Checking.	
Unit-IV	**09 Lecture Hours**
Code Optimization: Code Optimization: Introduction, The Principal sources of optimization, Optimization of basic blocks, Loops in flow graphs, Introduction to global data-flow analysis, Iterative solution of data-flow equations, Code improving transformations, Dealing with aliases, Data-flow analysis of structured flow graphs, Efficient data-flow algorithms, A tool for data-flow analysis, Estimation of types, Symbolic debugging of optimized code.	
Unit-V	**10 Lecture Hours**
Run-Time Environment and Code Generation: Symbol tables: use and need of symbol tables. Runtime Environment: storage organization, stack allocation, access to non-local data, heap management, parameter passing mechanisms, introduction to garbage collection, Reference counting garbage collectors. Code generation: Issues, target language, Basic blocks & flow graphs, Simple code generator, Peephole optimization, Register allocation and assignment.	

Text Books:

Compilers, Principles Techniques and Tools- Alfred V Aho, Monica S Lam, Ravi Sethi, Jeffrey D. Ullman,2ndEdition, Pearson,2007.
Compiler construction, Principles and Practice, Kenneth C Louden, 1st Edition, CENGAGE **Reference Books:**
Compiler Design, K. Muneeswaran, Oxford. 2012
Engineering a compiler, Keith D.Cooper& Linda Torczon, Morgan Kaufman, 2nd edition. MK publishers, 2012.
Principles of compiler design, V. Raghavan, 2nd Edition, Tata Mcgraw Hill, 2011.

	Artificial Intelligence and Machine Learning (Elective -1)	L	T	P	C
Version 1.0	Contact hours -45	3	0	0	3
Pre-requisites/Exposure	Basics of Algorithm, Linear Algebra, Probability, and Statistics				
Co-requisites	--				

Course Objectives:

- To help the student to acquire knowledge of basics of artificial intelligent computing.
- To enable students to gain basic knowledge of machine learning.
- To incorporate the evolutionary computational knowledge.
- To enable students to acquire various problem solving, learning, and planning ability.
- To enable students to apply machine learning models to solve real-life problems.

Course Outcomes:

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

- CO1. Define solution according to real problem, apply search proper strategies for a particular problem, and construct logical propositions to conclude a proof statement.
- CO2. Construct and differentiate plan for specific problem solution using various planning strategies.
- CO3. Implement predictive and classification model using regression method.
- CO4. Design and deploy Multilayer Artificial Neural Network using backpropagation algorithm for different dataset, probabilistic model using conditional probability (Baye's Theorem).
- CO5. Construct SVM for linearly and non-linearly (kernel method) separable data. Generate Ability to select best features from the dataset using PCA.

Catalogue Description:

There is a growing need for talented machine learning/data scientist developers across every industry. As technology advances, the ability to build quality machine learning driven software while considering design, development, security, and maintenance is sought after amongst all kinds of companies, from finance and banking to healthcare and national security.

Machine Learning applies the knowledge and theoretical understanding gained through computer science to building high-quality intelligent software products. As a maturing discipline, Artificial Intelligence is becoming more and more important in our everyday lives. Our software development and engineering professional program is University's response to the tremendous growth of the software development industry.

Course Content:

Unit I: **8 lecture hours**

Module 1:
Introduction, Agents, Problem formulation, Uninformed search strategies, Heuristics, Informed search strategies, Satisfying constraints

Logical agents, Propositional logic, Inference rules, First-order logic, Inferences in first order logic

Unit II: 10 lecture hours

Planning with state-space search, Partial-order planning, planning graphs, Planning and acting in the real world Forward and backward chaining, Unification, Resolution.

Introduction to Machine Learning: Overview of machine learning, related areas, applications, software tools, course objectives.

Basics of Machine Learning: Learning Topologies: Training-Testing-Validation; Error: Actual Output; Target Output; Error Optimization: Gradient Descent (SGD, Minibatch); Parameter Update; Dataset and cleaning, Normalization; Bias and Variance; Hypothesis Testing;

Unit III: 6 lecture hours

Regression: Linear Regression: Single, Polynomial Regression, Gradient Descent, ANOVA, Logistic Regression, Generalization: Ridge and Lasso regression.

Case Study: Media Company Case Study; Cynlate Bank Loan Disbursement.

Unit IV: 11 lecture hours

Neural networks: The perceptron algorithm, various activation functions and their differentiability, multilayer perceptrons, back-propagation, nonlinear regression, multiclass discrimination, training procedures, Bayesian

Learning, Decision Tree

Unsupervised Learning: Uses of Unsupervised Learning; Data Clustering: K-means and Kernel K-means;

Unit V: 10 lecture hours

Support vector machines: Functional and geometric margins, optimum margin classifier, constrained optimization, Lagrange multipliers, KKT conditions, soft margins, kernels.
Dimensionality Reduction: Feature Selection, Principle Component Analysis (PCA).

Text Books:

1. Artificial Intelligence – A Modern Approach, Second Edition, S. Russel and P. Norvig Pearson Education, 2003.
2. Artificial Intelligence, Ritch& Knight, TMH
3. "Machine Learning", 1st Edition, Tom M. Mitchell, McGraw-Hill Series In Computer Science

Reference Books:

1. Artificial Intelligence; Structures for Complex Problem Solving, Fourth edition, G. Lugar, Pearson Education, 2002
2. Artificial Intelligence: A New Synthesis, Nils J. Nilsson, Morgan Kaufmann Publishing, Inc., Year 1998
3. "INTRODUCTION TO MACHINE LEARNING", 2005 Edition, Nils J Nillsson, Morgan Kaufmann
4. "Foundations of Machine Learning", 2012 Edition, Mehryar Mohri, Afshin Rostamezadeh, Ameet Talwalkar, The MIT Press
5. Python Data Science Handbook Essential Tools for Working with Data", 1st Edition, Jake Vander Plas, O'Reilly

	Fundamental of Cloud Computing (Elective-1)	**L**	**T**	**P**	**C**
Version 1.0	**Contact Hours -45**	**3**	**0**	**0**	**3**
Pre-requisites/Exposure	**DBMS, Java, Python, Computer Networking**				
Co-requisites	--				

Course Objectives:

- To introduce cloud computing-based programming techniques and cloud services.
- To introduce concepts and security issues of cloud paradigm.
- To impart the fundamentals of virtualization techniques.

Course Outcomes:

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

- CO1. How to provide Flexible and scalable infrastructures.
- CO2. Organize process to reduce implementation and maintenance costs.
- CO3. The case studies will help us to understand more of practice of cloud computing in the market.
- CO4. Determine flexible and scalable infrastructure suitable to the organizational need.
- CO5. Comparison of cost-wise solution to the problem and selecting the best solution for the problem suggested to the organization.

Catalogue Description:

This course focuses on concepts of cloud, fundamental building blocks like Resource Consolidation, Hypervisor, VM etc. and the cloud service models. It gives students the insight into how to build clouds. And provides practices on building the cloud. It also gives exposure to Public and Privacy Clouds. It gives students the future directions in cloud domain.

Course Content:

Unit I: 08 lecture hours

Data communication Components: Overview, Roots of Cloud Computing, Layers and Types of Cloud, Desired Features of a Cloud, Benefits and Disadvantages of Cloud Computing, Cloud Infrastructure Management, Infrastructure as a Service Providers, Platform as a Service Providers, Challenges and Risks.

Unit II: 10 lecture hours

Working with Cloud- Infrastructure as a Service: conceptual model and working Platform as a Service: conceptual model and functionalities Software as a Service: conceptual model and working Technologies and Trends in Service provisioning with clouds.

Service management, Computing on demand, Identity as a Service, Compliance as a Service

Unit III: 6 lecture hours

Abstraction and Virtualization: Introduction to Virtualization Technologies, Load Balancing and Virtualization, Understanding Hyper visors, Understanding Machine Imaging, Porting Applications, Virtual Machines Provisioning and Manageability, Virtual Machines Manageability, Virtual Machine Migration Services, Virtual Machine Provisioning and Migration in Action, Provisioning in the Cloud Context.

Unit IV: 10 lecture hours

Cloud Infrastructure and Cloud Resource: Management Architectural Design of Compute and Storage Clouds, Layered Cloud Architecture Development, Design Challenges, Inter Cloud Resource Management, Resource Provisioning and Platform Deployment, Global Exchange of Cloud Resources., Administrating the Cloud, Cloud Management Products, Emerging Cloud Management Standards.

Unit V: 11 lecture hours

Cloud Security: Security Overview, Cloud Security Challenges and Risks, Software-as-a Service Security, Cloud computing security architecture: Architectural Considerations, General Issues Securing the Cloud, Securing Data, Data Security, Application Security, Virtual Machine Security, Identity and Presence, Identity Management and Access Control, Autonomic Security, Storage Area Networks, Disaster Recovery in Clouds.

Text Books:

1. Rajkumar Buyya et. el., Cloud Computing: Principles and Paradigms, Wiley India Edition 2.
Sosinsky B., “Cloud Computing Bible”, Wiley India

Reference Books:

1. Mastering Cloud Computing by Rajkumar Buyya, C. Vecchiola & S. Thamarai SelviMcGRAW Hill Publication
2. Miller Michael, “Cloud Computing: Web Based Applications that Change the Way You Work and Collaborate Online”, Pearson Education India
3. Velte T., Velte A., Elsenpeter R., “Cloud Computing – A practical Approach”, Tata McGrawHill

	Artificial Intelligence and Machine Learning Lab	**L**	**T**	**P**	**C**
Version 1.0	**Contact hours -30**	**0**	**0**	**4**	**2**
Pre-requisites/Exposure	**Data Structure and Python Basics**				

Co-requisites	--

Course Objectives:

- To help students gain practical insights of AI Algorithm through functional programming.
- To enable students, communicate with clarity and precision of ML Algorithm.
- To give the students a perspective enhancement of present system.
- To enable students to make a comparative study and further improvement.

Course Outcomes:

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

- Implement and Evaluate different search strategies using Prolog.
- Execute and Memorize and different various libraries and the most frequently used functions, methods, constants required for the implementation of any machine learning.
- Implement and Appraise Linear and Logistic Regression and Classify using K-NN for smaller dataset.
- Implement clustering algorithm and judge the appropriate clustering method for a particular dataset.
- Also, to design Artificial Neural Network for different dataset and to classify for multiclass datasets. Implement Decision Tree and Naïve Bayes classifier, Design Linear SVM and appraise.

Catalogue Description:

Every laboratory course brings an open world to a student. It helps the most in exploring and innovating. In Artificial Intelligence and Machine Learning Lab all experiments are given based on real-life problems. Through this kind of practice students become more analytic more inclined to practical thinking. Also, this course brings inquisitiveness to the students. This course is a rationale to the advance courses such as Artificial Neural Network and Deep Learning", "Soft Computing", etc. First Part of this course is the implementation of some important Artificial intelligence aspects such as Agents, Knowledge Representation and Planning. The later part implements all major Machine Learning algorithms with the online datasets.

Course Content:

Experiments:

1. Introduction to TensorFlow:

2. Introduction;
3. Installation; Introduction to Tensors – Variable, Constants; Data Flow Graph; TensorBoard; Mathematics with TensorFlow.
4. Starting with Machine Learning using TensorFlow:
5. Linear Regression (Lasso and Ridge Regression, Elastic Net Regression)
6. Dataset loading
7. Gradient Descent Algorithm
8. Accuracy
9. Optimization
10. ROC and AUROC Curve generation
11. Nearest Neighbour Model (K-NN) Classifier
12. Data Clustering (K-Means, K-Medoid, Hierarchical Clustering)
13. Artificial Neural Network
14. Introducing Neural Networks
15. TensorFlow implementation of Single Layer Perceptron using logistic regression
16. Building the model
17. Fit the model
18. Test evaluation
19. TensorFlow implementation of Multi-Layer Perceptron
20. Multi-Layer Perceptron classification
21. Build the model
22. Fit the model
23. Test evaluation
24. Multi-Layer Perceptron function approximation
25. Build the model
26. Fit the model
27. Test evaluation
28. Bayesian Method implementation using TensorFlow
29. Decision Tree
30. Implementation of ID3 algorithm using Tensorflow
31. Support Vector Machine (SVM)
32. Linear SVM
33. Reduction to Linear Regression
34. SVM Kernels in TensorFlow
35. Implementation of a Non-Linear SVM
36. Implementation of a Multi-Class SVM
37. Principle Component Analysis using TensorFlow
38. Linear Dependence and Span
39. Norms

40. Special Kinds of Matrices and Vectors
41. Eigen decomposition
42. Singular Value Decomposition
43. The Moore-Penrose Pseudoinverse
44. The Trace Operator.
45. The Determinant

Text Books:

G. Zaccone, Getting started with TensorFlow: Get up and running with the latest numerical computing library by google and dive deeper into your data! Community experience distilled, Packt Publishing, 2016.

A. G´eron, Hands-on machine learning with SciKit-Learn, Keras, And TensorFlow: Concepts, tools, and techniques to build intelligent systems, O'Reilly Media, Incorporated, 2019.

Reference Books:

"Python Data Science HandbookEssential Tools for Working with Data", 1st Edition, Jake VanderPlas, O'Reilly

	Foundation of Cloud Computing Lab	**L**	**T**	**P**	**C**
Version 1.0	**Contact Hours -45**	**0**	**0**	**4**	**2**
Pre-requisites/Exposure	**DBMS, Java, Python**				
Co-requisites	**--**				

Course Objectives:

- To understand the installation of hypervisors.
- To understand the installation of different cloud simulation tools and cloud setup tools.
- To deploy cloud services.

Course Outcomes:

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

- CO1. Describe the key concepts and technologies in cloud computing.
- CO2. Evaluate cloud computing technologies and platforms in the context of the needs of a specific application
- CO3. Design data storage components for cloud-based software systems.
- CO4. Assess and monitor resource use of applications in virtualized environments
- CO5. Design, implement, and deploy cloud applications for current cloud platforms
- CO6. Evaluate privacy and security issues for cloud infrastructure and cloud applications

Catalogue Description:

This course introduces students to fundamentals of cloud computing and software development for cloud platforms. It covers topics such as virtualization, architecture of cloud systems, programming for the cloud, resource management, as well as privacy and security issues. Students gain practical experience developing applications for cloud platforms through a series of hands-on assignments.

Course Content:

Experiment 1: Introduction to cloud computing
Experiment 2: Hands on creation of virtual machine using computer server.
Experiment 3: Design virtual machine
Experiment 4: Key based authentication and login virtual machine from the host machine
Experiment 5: Create Backend logic to communication with frontend app using Ajax
Experiment 6: Using Backend logic setup communication with frontend app using Ajax
Experiment 7: Create SQL DB and design schema for user session 265. Login using username and password and validate in SQL

Experiment 8: Procedure to setup one Hadoop Cluster Access the Hadoop using API's from the application and show the data
Experiment 9: Demonstrate the use of map/reduce using simple program 269.
AWS Free Tier Account Creation
Experiment 10: In AWS account enabling Multi-Factor Authentication to Secure Your Access and create your First Linux Instance
In AWS create your First EC2 windows instance In AWS assign Elastic IP Addresses to Instance (Static IP Address)

Text Books:

1. Barrie Sosinsky, "Cloud Computing Bible", Wiley India Edition.
2. Anthony Velte, tobyVelte, Robert Elsenpeter, "Cloud Computing – A Practical Approach", Tata McGraw-Hill Edition.

	Natural Language Processing and Its Application (Elective -II)	**L**	**T**	**P**	**C**
Version 1.0	**Contact Hours -45**	**3**	**0**	**0**	**3**
Pre-requisites/Exposure	**Computer Programming with python, Computer Programming with Python Lab**				
Co-requisites	**-**				

Course Objectives:

- To understand key concepts from NLP are used to describe and analyse language
- To understand semantics and pragmatics of language for processing
- To apply structured semantic models on information retrieval and natural language applications.

Course Outcomes:

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

- CO1. Recall linguistic phenomena and an ability to model them with formal grammars.
- CO2. Illustrate proper experimental methodology for training and evaluating empirical NLP systems
- CO3. Identify natural language processing techniques to process speech and analyse text.
- CO4. Examine algorithms of natural language processing CO5. Evaluate different language modeling Techniques.

Catalogue Description:

This course introduces the fundamental concepts and techniques of natural language processing (NLP). Students will gain an in-depth understanding of the computational properties of natural languages and the commonly used algorithms for processing linguistic information. The course examines NLP models and algorithms using both the traditional symbolic and the more recent statistical approaches.

Course Content:

Unit I: 5 lecture hours

Introduction: Context - Classical Toolkit - Text Pre-processing – Tokenization – Sentence Segmentation

Lexical Analysis: Finite State Morphonology Paradigm based Lexical Analysis - Syntactic Parsing – CockeKasami-Younger Algorithm – Deductive Parsing – LR Parsing – Constraint based Grammars – Issues in Parsing Semantic Analysis: Theories and approaches to Semantic Representation – Fine Grained Lexical

Case studies - Natural Language Generation – Components of a Generator – Approaches to Text Planning – Linguistic Component.

Unit II: 11 lecture hours

Corpus Size, Representation, Sampling – Data Capture – Corpus Markup and Annotation – Multilingual Corpora – Multimodal Corpora -Corpus Annotation Types

Morphosyntactic Annotation – Treebanks: Syntactic, Semantic, and Discourse Annotation - Process of Building Treebanks - Applications of Treebanks - Searching Treebanks.

Fundamental Statistical Techniques: Binary Linear Classification – One-versus-All Method for MultiCategory Classification - Maximum Likelihood Estimation - Generative and Discriminative Models - Mixture Model and EM - Sequence Prediction Models.

Part-of-Speech Tagging: General Framework – POS Tagging Approaches – Other Statistical and Machine Learning Approaches.

Statistical Parsing: Basics - Probabilistic Context-Free Grammars - Generative Models – Discriminative Models
- Beyond Supervised Parsing.

Unit III: 10 lecture hours

Multiword Expressions (MWE): Linguistic Properties, Types, Classification of MWEs – Research Issues

Methods of Word Similarity – Normalized Web Distance Method – Kolmogorov Complexity – Information Distance – Normalized Web Distance – Applications – Word Sense Inventories and Problem Characteristics – Applications of Word Sense Disambiguation – Approaches to Sense Disambiguation: Supervised, Lightly Supervised and Unsupervised.

Unit IV: 10 lecture hours

Modern Speech Recognition: Architectural Components – Historical Developments – Speech Recognition Applications – Technical Challenges and Future Research Directions

Alignment: Basics – Sentence Alignment – Character, Word, Phrase Alignment – Structure and Tree Alignment – Biparsing and ITG Tree Alignment

Statistical Machine Translation: Approaches – Language Models – Parallel Corpora – Word Alignment – Phrase Library – Translation Models – Search Strategies – Research Areas.

Unit V: 09 lecture hours

Information Retrieval – Indexing – IR Models – Evaluation and Failure Analysis

Natural Language Processing and Information Retrieval – Question Answering – Generic Question Answering System – Evaluation of Question Answering system – Multilingualism in Question Answering System

Recent trends and Related Works – Information Extraction – IE with Cascaded Finite State Transducers – Learning based Approaches in IE – Report generation – Emerging Applications of Natural language Generation in Information – Biomedical Text Mining – Sentiment Analysis and Subjectivity.

Text Books:

1. Daniel Jurafsky and James H. Martin Speech and Language Processing (2nd Edition), Prentice Hall; 2 editions, 2008

2. Foundations of Statistical Natural Language Processing by Christopher D. Manning and Hinrich Schuetze, MIT Press, 1999

Reference Books:

1. James Allen, Natural Language Understanding, Addison Wesley; 2 edition 1994

2. Steven Bird, Ewan Klein and Edward Loper Natural Language Processing with Python, O'Reilly Media; 1 edition, 2009

	Data Warehousing & Data Analytics (Elective -II)	**L**	**T**	**P**	**C**
Version 1.0	**Contact Hours -45**	**3**	**0**	**0**	**3**
Pre-requisites/Exposure	**DBMS, Java, Python, Computer Networking**				
Co-requisites	--				

Course Objectives:

- To be familiar with mathematical foundations of data mining tools.
- To understand and implement classical models and algorithms in data warehouses and data mining
- To characterize the kinds of patterns that can be discovered by association rule mining, classification and clustering.
- To understand the Big Data Platform and its Use cases
- To provide an overview of Apache Hadoop
- To provide HDFS Concepts and Interfacing with HDFS
- To understand Map Reduce Jobs

Course Outcomes:

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

- CO1. Understand the functionality of the various data mining and data warehousing component.
- CO2. Appreciate the strengths and limitations of various data mining and data warehousing models.
- CO3. Explain the analysing techniques of various data.
- CO4. Identify Big Data and its Business Implications.
- CO5. List the components of Hadoop and Hadoop Eco-System

Catalogue Description:

This course will introduce the concepts of data ware house and data mining, which gives a complete description about the principles, used, architectures, applications, design and implementation of data mining and data ware housing concepts along with the added advantage of data analytics. Which will familiarize students with big data analysis as a tool for addressing substantive research questions. The course begins with a basic introduction to big data and discusses what the analysis of these data entails, as well as associated technical, conceptual and ethical challenges. Strength and limitations of big data research are discussed in depth using real-world examples.

Course Content:

Unit I: 09 lecture hours

Data Warehouse Fundamentals: Introduction to Data Warehouse, Data Warehouse Environment, OLTP Systems; Differences between OLTP Systems and Data Warehouse: Characteristics of Data Warehouse; Functionality of Data Warehouse: Advantages and Applications of Data Warehouse; Advantages, Applications Planning and Requirements: Introduction: Planning Data Warehouse and Key Issues: Planning and Project Management in constructing Datawarehouse: Data Warehouse Project; Data Warehouse development Life Cycle, Kimball Lifecycle Diagram, Requirements Gathering Approaches: Team organization, Roles, and Responsibilities:

Unit II: 6 lecture hours

Data Warehouse Architecture: Introductions, Components of Data warehouse Architecture: Data Warehouse vs Data Mart, Component based architecture, Dimensional Modelling: Introduction: E-R Modelling: Dimensional Modelling: E-R Modelling VS Dimensional Modelling: Data Warehouse Schemas; Star Schema, Inside Dimensional Table.

Unit III: 8 lecture hours

Bring data into data warehouse: ETL, Role of data transformation, Data Warehouse & OLAP: Introduction: What is OLAP? Characteristics of OLAP, Steps in the OLAP Creation Process, Advantageous of OLAP: What is Multidimensional Data: OLAP Architectures; MOLAP, ROLAP, HOLAP.

Unit IV: 10 lecture hours

Data pre-processing : Data cleaning , Data transformation , Data mining knowledge representation, Data Mining Algorithms: Association Rules, Classification, Prediction Data Definitions and Analysis Techniques Elements, Variables, and Data categorization Levels of Measurement Data management and indexing Introduction to statistical learning Descriptive Statistics: Measures of central tendency, Measures of location of dispersions, Basics and Usage of Rapidminer and Weka as mining tool.

Unit V: 12 lecture hours

Basic analysis techniques Statistical hypothesis generation and testing Chi-Square test t-Test Analysis of variance Correlation analysis Maximum likelihood test. Foundation of Data Analytics: - Introduction ,Evolution
, Concept and Scopes , Data , Metrics and Data classification, Data Reliability & Validity, Problem Solving with Analytics, Different phases of Analytics in the business and Data science domain, Descriptive Analytics, Predictive Analytics and Prescriptive Analytics , Different Applications of Analytics in Business, Data analysis techniques: Regression analysis; Classification techniques; Clustering: Association rules analysis. Case studies and projects: Understanding business scenarios; Feature engineering and visualization; Case Studies: Finance, Car Sales, Bank Customer Complaints, etc.

Text Books:

1. Data Ware Housing Fundamentals, Pualraj Ponnaiah, Wiley Student Edition
2. Data Mining-Concepts and Techniques- Jiawei Han, Micheline Kamber, Morgan Kaufmann

Publishers, Elsevier, 2 Edition, 2006.

3. Data Warehousing, Data Mining, and OLAP, Alex Berson, Stephen J. Smith, MGH, 1998.
4. Data Analytics: Become A Master In Data Analytics, Richard Dorsay, CreateSpace Independent Publishing Platform.

Reference Books:

1. The Data Ware House Life Cycle Toolkit- Ralph Kimball, Wiley Student Edition.
2. Data Mining Introductory and advanced topics –MARGARET H DUNHAM, PEARSON EDUCATION

	Public Blockchain	**L**	**T**	**P**	**C**
Version 1.2	**Contact Hours – 45**	**3**	**0**	**0**	**3**
Pre-requisites/Exposure	**Crypto currency and computer security basics**				
Co-requisites	-				

Course Objectives:

- To gain knowledge about the building blocks of blockchain ethereum.
- To enable students to install and configure Mist browser,
- To give the students a perspective to learn the basics of EVM and Solidity programming.
- To enable students acquire knowledge about smart contract and tokens.

Course Outcomes:

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

- CO1. Understand the basics of blockchain ethereum.
- CO2. Explain the procedure of installation of Mist browser and its configuration.
- CO3. Explain the role of Ethereum protocol in Banking.
- CO4. Understand the basics of Solidity programming primer.
- CO5. Understand the utility of smart contract and token.
- CO6. Evaluate the ancestry of blocks and transactions.

Catalogue Description:

This course is the definitive introduction to permissioned blockchain for the students. Beyond the technology, this course will introduce you to some of the philosophy behind decentralization and why there is so much excitement around it.
During the tenure of the course, the students will be introduced to blockchain and the technology behind it. In the later modules, the topics beyond bitcoin will be taken up and delve deeper into a next-generation blockchain called Ethereum to introduce students to what modern blockchains can do.

Course Content:

Unit I: **6 lecture hours**

Bridging the Blockchain Knowledge Gap: What Ethereum Does, Three Parts of a Blockchain, Ether as a Currency and Commodity, The Power Is in the Protocol, You Can Build Trustless Systems, What Smart Contracts: Objects and Methods for Value, Just Add Commerce, Content Creation; Where's the Data? :What Is Mining, Ether and Electricity Prices; EVM: The Mist Browser, Browser vs. Wallet or Keychain;What Ethereum Is Good For: State of Smart Contract Development

Unit II: 8 lecture hours

The Mist Browser: introduction, The Bank Teller Metaphor, In Cryptocurrency, You Hold Your Own Assets, Visualizing Ethereum Transactions, Breaking with Banking History, How Encryption Leads to Trust, System Requirements, more about Eth.guide and This Book, Tools for Developers, CLI Nodes, Recommended: Using Parity with Geth, Finally, into the Mist! , Downloading and Installing Mist, Configuring Mist, Finding Your New Address, Sending and Receiving Ether, Understanding Ethereum Account Types, Backing Up and Restoring Your Keys, Using Paper Wallets, Using Mobile Wallets, Working with Messages and Transactions, So, What Is a Blockchain? , Paying for Transactions, Understanding Denominations, Getting Ether, Anonymity in Cryptocurrency, Blockchain Explorers .

Unit III: 8 lecture hours

The EVM: The Role of the Ethereum Protocol in Banking, What the EVM Does, EVM Applications Are Called

Smart Contracts, The Name "Smart Contracts, Understanding State Machines, Digital vs. Analog, "Statements", Data's Role in State, How the Guts of the EVM Work, The EVM Constantly Checks for Transactions, creating a Common Machine Narrative of What Happened, Cryptographic Hashing, Hash Functions (or Hash Algorithms),
Blocks: The History of State Changes, Block Time, Drawbacks of Short Blocks, "Solo Node" Blockchain, Distributed Security, Mining's Place in the State Transition Function, Renting Time on the EVM, Gas: Why Is Gas So Important, Why Isn't Gas Priced in Ether, Fees as Regulation, Working with Gas

Unit IV: 8 lecture hours

Solidity Programming Primer: Global Banking Made (Almost) Real, Extra-Large Infrastructure, Worldwide
Currency, Complementary Currency, The Promise of Solidity, Browser Compiler, Learning to Program the EVM, Easy Deployment, The Case for Writing Business Logic in Solidity, Code, Deploy, Relax, Design Rationale, Writing Loops in Solidity, Expressiveness and Security, The Importance of Formal Proofs, Historical Impact of a Shared Global Resource, How Attackers Bring Down Communities, Hypothetical Attack Written in Solidity, Automated Proofs to the Rescue, Determinism in Practice, Lost in Translation,

Unit V: 7 lecture hours

Smart Contracts and Tokens: EVM as Back End, Smart Contracts to Dapps, Assets Backed by Anything, Bartering with Fiat Currency, Ether as Glass Beads, Cryptocurrency Is a Measure of Time, Asset Ownership and Civilization, Coins are Collectibles, The Function of Collectibles in Human Systems, Early Counterfeiting, Jewellery and Art as Money , The Step Toward Banknotes , Platforms for High-Value Digital Collectibles , Tokens Are a Category of Smart Contract , Tokens as Social Contracts, Tokens Are a Great First App, Creating a Token on the Testnet , Getting Test Ether from the Faucet, Registering Your Tokens , Deploying Your First Contract, Same House, Different Address , Playing with Contracts .

Unit VI: 8 lecture hours

Mining Ether: Ether's Source, Mining, Self-Regulation, and the Race for Profit, How Proof of Work Helps Regulate Block Time, What's Going on with the DAG and Nonce, Making Fast Blocks Work, How Ethereum Uses Stale Blocks, Uncle Rules and Rewards, The Difficulty Bomb, Miner's Winning Payout Structure, Limits on Ancestry, The Block Processing Play by Play, Evaluating the Ancestry of Blocks and Transactions, How Ethereum and Bitcoin Use Trees, Merkle-Patricia Trees, Contents of an Ethereum Block Header, Forking, Installing Geth on macOS

Text Books:

1. Mayukh Mukhopadhyay - Ethereum smart contract development_ build blockchain-based decentralized applications using Solidity-Packt Publishing (2018)

2. Chris Dannen (auth.) - Introducing Ethereum and Solidity_ Foundations of Cryptocurrency and Blockchain Programming for Beginners-Apress (2017)

3. Mastering Bitcoin: Programming The Open Blockchain, Andreas M. Antonopoulos, O'Reilly, ISBN: 9789352135745.

Reference Books:

1. Ethereum for Architects and Developers: With Case Studies and Code Samples in Solidity by Debajani Mohanty

2. Blockchain for Business by Jai Singh Arun

3. Blockchain Applications: A Hands-on Approach by Arshdeep Bahga and Vijay K. Madisetti,
ISBN: 9780996025560.