School of Engineering and Applied Sciences
B. Tech Computer Science and Engineering

Academic Batch: 2020-2024

Department of Computer Science Engineering
SRM University-AP, Andhra Pradesh

B. Tech Curriculum in Computer Science Engineering

Semester-I						
S. No	**Course Code**	**Course Name**	**L**	**T**	**P**	**C**
1	EGL 101	Communicative English	3	0	0	3
2	MAT 112	Single Variable Calculus	3	0	0	3
3	PHY 101/ CHE 101	Engineering Physics / Principles of Chemistry	3/2	0	0	3/2
4	PHY 101 L/ CHE 101 L	Engineering Physics Lab / Principles of Chemistry Lab	0	0	2	1
5	BIO 102 / ENV 111	Introductory Biology / Environmental Science	3/2	0	0	3/2
6	***/ENV 111 L	***/Environmental Science Lab	0/0	0/0	0/2	0/1
7	CSE 105	Introduction to Programming Using C	3	0	0	3
8	CSE 105 L	Introduction to Programming Using C Lab	0	0	2	1
9	ISES 101	Industry Specific Employability Skills-I	1	1	0	1
Total						**18/17**

Semester-II						
S. No	**Course Code**	**Course Name**	**L**	**T**	**P**	**C**
1	MAT 121	Multi variable Calculus	3	0	0	3
2	MAT 221	Probability and Statistics for Engineers	3	0	0	3
3	ENG 111	Basic Electronics	3	0	0	3
4	ENG 111 L	Basic Electronics Lab	0	0	2	1
5	CSE 107	Data Structures	3	0	0	3
6	CSE 107 L	Data Structures Lab	0	0	2	1
7	CHE 101 / PHY 101	Principles of Chemistry / Engineering Physics	2/3	0	0	2/3
8	CHE 101 L/ PHY 101 L	Principles of Chemistry Lab / Engineering Physics Lab	0	0	2	1
9	BIO 102 / ENV 111	Introductory Biology / Environmental Science	2/3	0	0	2/3
10	***/ENV 111 L	***/Environmental Science Lab	0/0	0/0	2/0	1/0
11	ISES 102	Industry Specific Employability Skills-II	1	1	0	1
** No Lab for Biology						
Total						**21/22**

Semester-III						
S. No	**Course Code**	**Course Name**	**L**	**T**	**P**	**C**
1	MAT 141	Discrete Mathematics	3	0	0	3
2	CSE 206	Object Oriented Programming with C++	3	0	0	3
3	CSE 206 L	Object Oriented Programming with C++ Lab	0	0	2	1
4	CSE 201	Design and Analysis of Algorithms	3	0	0	3
5	CSE 201 L	Design and Analysis of Algorithms Lab	0	0	2	1
6	ECE 211	Digital Electronics	2	1	0	3
7	ECE 211 L	Digital Electronics Lab	0	0	2	1
8	CSE 106 L	Hands on Using Python	0	0	4	2
9	CSE 230	Industry Standard Coding Practice-1	0	0	4	1
10	ECO 121	Principles of Economics	3	0	0	3
11	ISES 201	Industry Specific Employability Skills-III	1	1	0	1
Total						**22**

Semester-IV						
S. No	**Course Code**	**Course Name**	**L**	**T**	**P**	**C**
1	HSE	Humanities Elective	3	0	0	3
2	MAT 131	Differential Equations	3	0	0	3
3	CSE 204	Computer Organization and Architecture	3	0	0	3
4	CSE 204 L	Computer Organization and Architecture Lab	0	0	2	1
5	CSE 301	Operating System	3	0	0	3
6	CSEC 301 L	Operating System Lab	0	0	2	1
7	CSE 207	Java Programming	3	0	0	3
8	CSE 207 L	Java Programming Lab	0	0	2	1
9	CSE 203	Formal Languages and Automata Theory	3	0	0	3
10	ISES 202	Industry Specific Employability Skills-IV	1	1	0	1
11	CSE 330	Industry Standard Coding Practice-2	0	0	4	1
Total						**23**

Semester-V						
S. No	**Course Code**	**Course Name**	**L**	**T**	**P**	**C**
1	MAT 211	Linear Algebra	3	0	0	3
2	CSE 303	Computer Networks	3	0	0	3

3	CSE 303 L	Computer Networks Lab	0	0	2	1
4	CSE 306	Compiler Design	3	0	0	3
5	CSE 306 L	Compiler Design Lab	0	0	2	1
6	CSE 304	Database Management System	3	0	0	3
7	CSE 304 L	Database Management System Lab	0	0	2	1
8	CSE SE 1	CS Stream Elective 1	3	0	0	3
9	CSE SE 1 L	CS Stream Elective 1 Lab	0	0	2	1
10	OE	Open Elective 1	3	0	0	3
11	CSE 331	Industry Standard Coding Practice-3	0	0	4	1
12	ISES 301	Industry Specific Employability Skills-V	1	1	0	0
13	CSE 340	UROP	0	0	6	3
Total						**26**

Semester-VI						
S. No	**Course Code**	**Course Name**	**L**	**T**	**P**	**C**
1	CSE 305	Software Engineering	3	0	0	3
2	CSE 305 L	Software Engineering-Lab	0	0	2	1
3	OE	Open Elective 2	3	0	0	3
4	OE	Open Elective 3	3	0	0	3
5	CSE TE 1	CSE Technical Elective 1	3	0	0	3
6	CSE SE 2	CS Stream Elective 2	3	0	0	3
7	CSE SE 2 L	CS Stream Elective 2 Lab	0	0	2	1
8	ISES 302	Industry Specific Employability Skills-VI	1	1	0	0
Total						**17**

Semester-VII						
S. No	**Course Code**	**Course Name**	**L**	**T**	**P**	**C**
1	CSE SE 3	CS Stream Elective 3	3	0	0	3
2	CSE SE 3 L	CS Stream Elective 3 Lab	0	0	2	1
3	CSE SE 4	CS Stream Elective 4	3	0	0	3
4	CSE SE 4 L	CS Stream Elective 4 Lab	0	0	2	1
5	CSE TE 2	CS Technical Elective 2	3	0	0	3
6	OE	Open Elective 4	3	0	0	3
7	OE	Open Elective 5	3	0	0	3
Total						**17**

Semester-VIII						
S. No	**Course Code**	**Course Name**	**L**	**T**	**P**	**C**
2	CSE 462	Capstone Project	0	0	30	15
Total						**15**

Course Category	**Category Code**	**No of Courses**	**Credits in Curriculum**
Humanities and Social Sciences	HS	9	13
Basic Sciences	BS	11	25
Engineering Sciences	ES	10	18
Professional Core	C	23	48
Professional Elective	SE	8	16
	TE	2	6
Open Elective	OE	5	15
Project	PR	2	18
	Total	**70**	**159**

List of Stream Specific Electives

Course Code	Course Name	L	T	P	C
Artificial Intelligence and Machine Learning Stream					
CSE 413	Artificial Intelligence	3	0	0	3
CSE 413 L	Artificial Intelligence Lab	0	0	2	1
CSE 336	Machine Learning	3	0	0	3
CSE 336 L	Machine Learning Lab	0	0	2	1
CSE 314	Digital Image Processing	3	0	0	3
CSE 314 L	Digital Image Processing Lab	0	0	2	1
CSE 412	Principles of Soft Computing	3	0	0	3
CSE 412 L	Principles of Soft Computing Lab	0	0	2	1
Cyber Security Stream					
CSE 337	Cryptography	3	0	0	3
CSE 337 L	Cryptography Lab	0	0	2	1
CSE 315	Network Security	3	0	0	3
CSE 315 L	Network Security Lab	0	0	2	1
CSE 410	Mobile and Wireless Security	3	0	0	3
CSE 410 L	Mobile and Wireless Security Lab	0	0	2	1
CSE 414	Internet Protocols and Networking	3	0	0	3
CSE 414 L	Internet Protocols and Networking Lab	0	0	2	1
Big Data Analytics Stream					
CSE 310	Data Warehousing and Mining	3	0	0	3
CSE 310 L	Data Warehousing and Mining Lab	0	0	2	1
CSE 338	Applied Data Science	3	0	0	3
CSE 338 L	Applied Data Science Lab	0	0	2	1
CSE 417	Principles of Big Data Management	3	0	0	3
CSE 417 L	Principles of Big Data Management Lab	0	0	2	1
CSE 419	Information Retrieval	3	0	0	3
CSE 419 L	Information Retrieval Lab	0	0	2	1
Distributed and Cloud Computing Stream					
CSE 316	Distributed Systems	3	0	0	3
CSE 316 L	Distributed Systems Lab	0	0	2	1
CSE 318	Cloud Computing	3	0	0	3
CSE 318 L	Cloud Computing Lab	0	0	2	1
CSE 416	Cloud Data Management	3	0	0	3
CSE 416 L	Cloud Data Management Lab	0	0	2	1
CSE 418	Service Oriented Computing	3	0	0	3
CSE 418 L	Service Oriented Computing Lab	0	0	2	1
Internet of Things Stream					

CSE 337	Cryptography	3	0	0	3
CSE 337 L	Cryptography Lab	0	0	2	1
CSE 318	Cloud Computing	3	0	0	3
CSE 318 L	Cloud Computing Lab	0	0	2	1
CSE 317	Embedded Systems	3	0	0	3
CSE 317 L	Embedded Systems Lab	0	0	2	1
CSE 319	IoT Design Protocols	3	0	0	3
CSE 319 L	IoT Design Protocols Lab	0	0	2	1

List of Technical Electives					
Course Code	**Course Name**	**L**	**T**	**P**	**C**
CSE 320	Web Programming	3	0	0	3
CSE 321	Human Computer Interaction	3	0	0	3
CSE 322	Advanced Computer Architecture	3	0	0	3
CSE 323	Natural Language Processing	3	0	0	3
CSE 324	Computer Graphics	3	0	0	3
CSE 325	Advanced Data Structures and Algorithms	3	0	0	3
CSE 326	Distributed Operating Systems	3	0	0	3
CSE 420	Data and Web Mining	3	0	0	3
CSE 421	Complexity Theory	3	0	0	3
CSE 422	Software Project Management	3	0	0	3
CSE 423	Multimedia	3	0	0	3
CSE 424	Deep Learning	3	0	0	3
CSE 425	Advanced Database Management Systems	3	0	0	3
CSE 426	Fog Computing	3	0	0	3
CSE 427	Parallel Algorithms	3	0	0	3
CSE 428	Web Services	3	0	0	3
CSE 429	Advances in Data Mining	3	0	0	3
CSE 327	Social Network Analysis	3	0	0	3
CSE 328	Recommender Systems	3	0	0	3

SEMESTER-I

SEMESTER-I

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
EGL 101	Communicative English	HS	3	0	0	3

UNIT I: RHETORIC AND PUBLIC SPEAKING
Rhetoric, Critical Thinking and Public Speaking; Thinking Outside the Box; How to Deliver a Speech; Fundamentals of Persuasion.

UNIT II: NON-VERBAL COMMUNICATION
Nonverbal Communication; Spatial distance, eye contact and appearances; How nonverbal communication is more important than words.

UNIT III: COMMUNICATION AND THE MEDIA
Persuasion and the media; Radio, television, film, social media and the internet; How the media sells ideas, images, products and lifestyles; Fundamentals of Informative/Scientific Speeches and Research; The Heart of the Speech – Powerful Narratives; The Power of Narrative.

UNIT IV: SMALL GROUP COMMUNICATION
Small group communication; Leadership, conflict and persuasion in groups; The importance of small groups in business; Dr. A. Fisher's Fundamentals of Small Groups; Group Problem Solving; Learning to say no – don't say you will when you won't, don't say yes and then don't do it, be true to your word.

UNIT V: PERSUASION, IDEOLOGY AND MEDIA BIAS.
Advanced Rhetoric; Ideology; Persuasive Fallacies; How to Construct a Persuasive Speech; How to Present Scientific Data in a Speech; Unmasking Media Bias and Ideology; Full circle – the dangers of rhetoric and ideology.

LIST OF PRACTICAL EXPERIMENTS

1	Writing Journal 1.
2	Essay on Images.
3	Oral Presentation.
4	Writing Journal 2.
5	Essay on Community.

.

TEXTBOOKS

1. Communication: Principles for a Lifetime. Beebe, Beebe and Ivy, 6th Edition, Pearson Publishing.

REFERENCES

1. Qualitative Communication Research Methods (2011) Bryan C. Taylor and Thomas R. Lindlof. Sage Publications, New Delhi, India, 3rd Edition.
2. The Fundamentals of Small Group Communication (2008) Scott A. Myers and Carolyn M. Anderson. Sage Publications, New Delhi, India.

SEMESTER-I

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
MAT 112	Single Variable Calculus	BS	3	0	0	3

UNIT I: SEQUENCES AND SERIES
Sequences, series, Sum of a series, Geometric series, p-series, Comparison test, root test, ratio test.

UNIT II: LIMITS AND CONTINUITY
Limit of a function at a point, one sided limits, Continuity, Limits involving infinity.

UNIT III: DIFFERENTIATION
Derivative at a point, Derivative as a function, Product Rule, Quotient Rule, Chain Rule, Implicit Differentiation, Rolle's Theorem, Mean Value Theorem.

UNIT IV: APPLICATIONS OF DERIVATIVES
Maxima and minima, Monotonic functions and first derivative test, Related rates, Concavity and curve sketching, Optimization problems, Newton's Method, Taylor and MacLaurin Series.

UNIT V: INTEGRATION
Area as a limit of finite sums, Definite and indefinite integral, Fundamental Theorem of Calculus, Integration by substitution and integration by parts, Area between curves, Arc length.

TEXTBOOKS

1. Thomas' Calculus, 14th Edition, (2018) – J. Hass, C. Heil, M. Weir, Pearson Education.
2. Introduction to Real Analysis, Fourth Edition (2014) – R. Bartle, D. Sherbert, John Wiley and Sons.

REFERENCES

1. Calculus and Analytic Geometry, Ninth Edition (2017) – G. Thomas, R. Finney, Addison Wesley.

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
PHY 101	Engineering Physics	BS	3	0	0	3

UNIT I: INTRODUCTION TO VECTOR ALGEBRA

Gradient, Divergence and curl and their physical significances, Gauss and Stokes theorems, Vector operators in different coordinate (Curvilinear, Cartesian, Cylindrical and spherical) systems.

UNIT II: ELECTROSTATICS

Coulomb's law, Gauss law, Electric field, Electrostatic Potential, Potential energy of system of charges Boundary Value problems, capacitance.

UNIT III: DIELECTRICS AND POLARIZATION

Electric dipole and dipole moment, Electric potential due to dipole, Electric field intensity due to dipole, Polarization P, Electric displacement D, Electric susceptibility and dielectric constant, Bound volume and surface charge densities, Electric field at an exterior and interior point of dielectric.

UNIT IV: MAGNETOSTATICS

Biot-Savart law, Ampere's law for force between two current carrying loops, Ampere's circuital law Equation of continuity, Energy density in magnetic field, magnetization of matter (B, H, M)
Magnetic susceptibility and permeability, Hysteresis loss, B-H curve, Diamagnetic, paramagnetic and ferromagnetic substances.

UNIT V: INTRODUCTION TO ELECTRODYNAMICS

Time varying fields: Faradays law of induction, generalization of Amperes' law, Maxwell's equation (Differential and Integral form), Wave equation and plane waves in free space.

TEXTBOOKS

1. MIT-- 8.02X online course material.
2. Introduction to Electrodynamics (4rd Edition) - David J. Griffiths (Publisher - PHI Learning, Eastern Economy Editions, 2012).
3. Electricity and Magnetism (Reprints 2007, 1st Edition 2001) A. S. Mahajan, A. A. Rangwala, (Publisher - McGraw-Hill Education).

REFERENCES

1. Electricity and magnetism Edward M Purcell, David J Morin, 3rd edition, Cambridge University, 2013.
2. Classical Electrodynamics (3rd Edition) - John David Jackson. (Publisher – Wiley).

SEMESTER-I / SEMESTER-II

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
PHY 101 L	Engineering Physics Lab	C	0	0	2	1

LIST OF PRACTICAL EXPERIMENTS

1. Revisions of Vernier caliper and Screw Gauge measurement methods.
2. Plotting experimental data in graphs and error analysis.
3. To determine the moment of inertia of a flywheel.
4. Measurement of time period for a given compound pendulum with different lengths.
5. To determine radius of gyration of a given pendulum.
6. Verification of Stefan`s Law.
7. Measurement of specific heat capacity of any given material.
8. Verify of Hooke's law and to determine spring contact for given spring combinations.
9. To determine the rigidity modulus of steel wire by torsional oscillations.
10. To calculate Young's modulus of a given material by deflection method.
11. To measure the capacitance as a function of area and distance between the plates.
12. To determine the dielectric constant of different dielectric materials.
13. Measurement of the induced voltage impulse as a function of the velocity of the magnet.
14. Calculation of the magnetic flux induced by a falling magnet as a function of the velocity of the magnet.
15. To study the magnetic field along the axis of a current carrying circular loop.
16. To study the dependency of magnetic field on the diameter of coil.
17. To investigate the spatial distribution of magnetic field between coils and determine the spacing for uniform magnetic field.
18. To demonstrate the superposition of the magnetic fields of the two individual coils.
19. Study of B-H-Curve To study permeability curve of a given material.

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
BIO 102	Introductory Biology	BS	3	0	0	3

UNIT I: BASIS OF LIFE AND DIVERSITY

Molecular evolution, Elements to molecules: water, carbohydrates, lipids, proteins, nucleic acids, vitamins and minerals. Diversity of life: virus, bacteria, archea and eukarya. Concept of terrestrial, aquatic and amphibians. Mode of energy & carbon utilization-auto, hetero and lithothrophs.

UNIT II: CELL BIOLOGY

Cell: morphology, cell organelles and functions. Concept of unicellular and multicellular organisms. Cell cycle and cell division: mitosis and meiosis. Basis of cell-cell communication and signaling.

UNIT III: MOLECULAR BIOLOGY

DNA and Chromosomes: structure and organization, DNA replication, Transcription, Translation. Introduction to genetic engineering.

UNIT IV: ENZYMES AND APPLICATIONS

Introduction to enzymes; classification, parameters influencing the enzyme activity, mechanism of enzyme action and enzyme inhibition. Commercial applications of microorganisms and enzymes.

UNIT V: BIOLOGICAL SEQUENCES AND DATABASES

DNA and Protein sequences, Concept of genomics, transcriptomics, proteomics and metabolomics. File formats of sequence storage: FASTA file, GenBank. Biological databases – NCBI and EMBL browsers, KEGG and UniProt databases. Usefulness of biological Metadata-Array expression and 1000 genomes. Application of BLAST and Protein/Gene ID conversion.

TEXTBOOKS

1. Thrives in Biochemistry and Molecular Biology, Edition 1, 2014, Cox, Harris, Pears, Oxford University Press.
2. Exploring Proteins, Ed. 1, 2014, Price and Nairn,Oxford University Press.
3. Thrives in Cell Biology, Ed. 1, 2013, Qiuyu Wang, Cris Smith and Davis, Oxford University Press.

REFERENCES

1. Cooper, G.M., Housman, R.E. The cell: a molecular approach. (2009).ASM Press, Washington D. C.
2. Lehninger, A. L., Nelson, D.L., &Cox, M. M. Lehninger principles of biochemistry. (2000). Worth Publishers, New York.
3. Wilson, K., Walker, Principle and techniques of biochemistry and molecular biology, (2005). 6thedn. Cambridge University Press, Cambridge.
4. Harvey Lodish, Arnold Berk and Chris A. Kaiser, Molecular Cell Biology, Ed. 8, 2016, W. H Freeman & Co (Sd).
5. Bruce Alberts, Alexander D. Johnson, Julian Lewis, David Morgan, Martin Raff, Keith Roberts, and Peter Walter. 2014. Molecular Biology of the Cell. (Sixth Edition). W. W. Norton & Company.

6. Scott Freeman, Kim Quillin, Lizabeth Allison, Michael Black, Emily Taylor, Greg Podgorski and Jeff Carmichael. 2016. Biological Science. (6th Edition). Pearson.
7. Bruce Alberts, Dennis Bray, Karen Hopkin, Alexander D. Johnson, Julian Lewis, Martin Raff, Keith Robert and Peter Walter. 2014. Essential Cell Biology. (4th Edition). W. W. Norton & Company.
8. Lisa A. Urry, Michael L. Cain, Steven A. Wasserman, Peter V. Minorsky, Jane B. Reece. 2016. Campbell Biology (11th Edition). Pearson.
9. Peter H Raven, George B Johnson, Kenneth A. Mason, Jonathan Losos and Susan Singer. 2016. Biology. (11th Edition). McGraw-Hill Education.

SEMESTER-I

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 105	Introduction to Programming Using C	ES	3	0	0	3

UNIT I: INTRODUCTION

Computer systems, hardware, and software. Problem solving: Algorithm / Pseudo code, flowchart, program development steps Computer languages: Machine, symbolic and high-level languages Creating and Running Programs: Writing, editing (any editor), compiling (gcc), linking, and executing in Linux environment Structure of a C program, identifiers Basic data types and sizes. Constants, Variables Arithmetic, relational and logical operators, increment and decrement operator's Conditional operator, assignment operator, expressions Type conversions, Conditional Expressions Precedence and order of evaluation, Sample Programs.

UNIT II

SELECTION & DECISION MAKING: if-else, null else, nested if, examples, Multi-way selection: switch, else-if, examples.

ITERATION: Loops - while, do-while and for, break, continue, initialization and updating, event and counter controlled loops and examples.

ARRAYS: Concepts, declaration, definition, storing and accessing elements, one dimensional, two dimensional and multidimensional arrays, array operations and examples. Character arrays and string manipulations.

UNIT III: MODULAR PROGRAMMING

Functions - Basics, parameter passing, storage classes extern, auto, register, static, scope rules, user defined functions, standard library functions, Passing 1-D arrays, 2-D arrays to functions. Recursive functions - Recursive solutions for fibonacci series, towers of hanoi. C Pre-processor and header files.

UNIT IV: POINTERS

Concepts, initialization of pointer variables, pointers as function arguments, passing by address, dangling memory, address arithmetic, character pointers and functions, pointers to pointers, pointers and multi-dimensional arrays, dynamic memory management functions, command line arguments.

UNIT V: ENUMERATED, STRUCTURE AND UNION TYPES

Structures - Declaration, definition, and initialization of structures, accessing structures, nested structures, arrays of structures, structures and functions, pointers to structures, self-referential structures. Unions, typedef, bit-fields, program applications. Bit-wise operators: logical, shift, rotation, masks.

FILE HANDLING: Concept of a file, text files and binary files, formatted I/O, file I/O operations and example programs.

TEXTBOOKS

1. The C programming Language by Brian Kernighan and Dennis Richie.

REFERENCES

1. Problem Solving and Program Design in C, Hanly, Koffman, 7th edition, PEARSON 2013.

2. Programming in C, Pradip Dey and Manas Ghosh, Second Edition, OXFORD Higher Education, 2011.
3. Programming in C, A practical approach Ajay Mittal PEARSON.
4. Programming in C, B. L. Juneja, Anith Seth, First Edition, Cengage Learning.

SEMESTER-I

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 105 L	Introduction to Programming Using C Lab	ES	0	0	2	1

LIST OR PRACTICAL EXPERIMENTS

Week-1: Basic C programs

a. Calculation of the area of triangle.
b. Swap two numbers without using a temporary variable.
c. Find the roots of a quadratic equation.
d. Takes two integer operands and one operator form the user, performs the operation and then prints the result.

1. **Week-2:** Loops
   a. Find the sum of individual digits of a positive integer and find the reverse of the given number.
   b. Generate the first n terms of Fibonacci sequence.
   c. Generate all the prime numbers between 1 and n, where n is a value supplied by the user.
   d. Print the multiplication table of a given number n up to a given value, where n is entered by the user.

2. **Week-3:** Loops
   a. Decimal number to binary conversion.
   b. Check whether the given number is Armstrong number or not.
   c. Triangle star patterns

I II III

3. **Week-4:** Arrays
   a. Interchange the largest and smallest numbers in the array.
   b. Searching an element in an array
   c. Sorting array elements.

4. **Week-5:** Matrix
   a. Transpose of a matrix.
   b. Addition and multiplication of 2 matrices.

5. **Week-6:** Functions

a. (nCr) and (nPr) of the given numbers
b. $1+x+x^2\backslash 2+x^3\backslash 3!+x^4\backslash 4!+\ldots\ldots\ldots..X^n\backslash n!$

6. **Week-7:** Functions and array
   a. Function to find both the largest and smallest number of an array of integers.
   b. Liner search.
   c. Replace a character of string either from beginning or ending or at a specified location.

7. **Weak-8**: Pre-processor directives
   a. If Def
   b. Undef
   c. Pragma

8. **Weak-9:** Structures
   a. Reading a complex number
   b. Writing a complex number.
   c. Addition of two complex numbers
   d. Multiplication of two complex numbers

9. **Weak-10:** String operations without using the built-in functions
   a. Concatenate two strings
   b. Append a string to another string.
   c. Compare two strings
   d. Length of a string
   e. Find whether a given string is palindrome or not

10. **Weak-11:** Pointers
    a. Illustrate call by value and call by reference.
    b. Reverse a string using pointers
    c. Compare two arrays using pointers

11. **Weak-12:** Pointers and array
    a. Array of Int and Char Pointers.
    b. Array with Malloc(), calloc() and realloc().

12. **Weak-13:** Recursion
    a. To find the factorial of a given integer.
    b. To find the GCD (greatest common divisor) of two given integers.
    c. Towers of Hanoi

13. **Weak-14:** File Operations
    a. File copy
    b. Word, line and character count in a file.

14. **Weak-15:** Command line arguments
    a. Merge two files using command line arguments.

SEMESTER-I

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
ISES 101	Industry Specific Employability Skills-I	HS	1	1	0	1

UNIT I: QUANTS

Speed calculations, Time and Distance, Problems on Trains, Boats and Streams, Races And Games, Escalator Problems, Time and Work , Chain Rule, Pipes and cistern, Simplification , surds and indices, Square roots and cube roots, Functions.

UNIT II: REASONING

Number Series, Alphabet series, Odd Man Out, Missing number, Wrong number, Analogies, Mathematical Operations, Calendars, Clocks, Cryptarithmetic, Identification of Cross-Variable Relation, Sudoku.

UNIT III: VERBAL

Basic sentence structure: Nouns, Pronouns, Adjectives. Parts of speech. Degree of comparison. Articles, conditionals, and sentences (kinds). Verb tense. Sentence formation. Paragraph formation, change of voice, Change of speech, Synonyms, Antonyms.

UNIT IV: COMMUNICATION SKILLS

Self-Introduction, Presentations, Email Etiquette

TEXTBOOKS/REFERENCE BOOKS/OTHER READING MATERIAL

1. Mitchell S. Green – 2017, Know Thyself: The Value and Limits of Self-Knowledge.
2. Debbie Hindle, Marta Vaciago Smith - 2013 , Personality Development: A Psychoanalytic Perspective.
3. Lani Arredondo - 2000, Communicating Effectively.
4. Patsy McCarthy, Caroline Hatcher - 2002, Presentation Skills: The Essential Guide for Students.
5. Martha Davis, Elizabeth Robbins Eshelman, Matthew McKay - 2008, Time Management and Goal Setting: The Relaxation and Stress.
6. Arun Sharma – How to prepare for Quantitative Aptitude, Tata Mcgraw Hill.
7. RsAgarwal,A Modern Approach to Verbal and Non Verbal Reasoning,S.Chand Publications.
8. Verbal Ability and Reading comprehension-Sharma and Upadhyay.
9. Charles Harrington Elstor, Verbal Advantage: Ten Easy Steps to a Powerful Vocabulary, Large Print, September 2000.
10. GRE Word List 3861 – GRE Words for High Verbal Score, 2016 Edition.
11. The Official Guide to the GRE-General Revised Test, 2nd Edition, Mc Graw Hill Publication.
12. English grammar and composition – S.C. Gupta.
13. R.S. Agarwal – Reasoning.
14. Reasoning for competitive exams – Agarwal.

SEMESTER-II

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
MAT 121	Multi Variable Calculus	BS	3	0	0	3

UNIT I: VECTORS AND MATRICES
Three-dimensional coordinate system, Vectors, Dot products, Vector products, Lines and planes.

UNIT II: PARTIAL DERIVATIVES
Functions of several variables, Limits and continuity for several variable functions, Partial derivatives, The chain rule, Directional derivatives, Gradient.

UNIT III: DOUBLE INTEGRAL ANDLINE, INTEGRAL IN PLANES
Extreme values, Saddle points, Lagrange multipliers.

UNIT IV: TRIPLE INTEGRALS IN 3D
Double and integrated integrals, Area by double integration.

UNIT V: SURFACE INTEGRALS IN 3D
Triple integration and applications.

TEXTBOOKS

1. Edwards, Henry C Thomas- Calculus, 14th edition. Chapters 12 to 16 relevant sections.
2. G.B. Thomas, Jr.and R. L. Finney, Calculus and Analytic Geometry, 9th Edn, Pearson Educations, India,1996.

REFERENCES

1. T. M. Apostol, Calculus - Vol.2, 2nd Edn., Wiley India, 2003.

SEMESTER-II

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
MAT 221	Probability and Statistics for Engineers	ES	3	0	0	3

UNIT I

Basic principle of counting, permutations, combinations, Multinomial coefficients, sample space and events, Axioms of probability, sample spaces having equally likely outcomes, Conditional probability, Bayes` theorem, independent events.

UNIT II

Random variable, discrete random variable, expected value, expectation of a function of a random, variable, variance, discrete probability distributions- Bernoulli, Binomial, Poisson, Geometric, negative, Binomial distributions, expected value of sums of random variables, cumulative distribution function and its properties.

UNIT III

Continuous random variables, Expectation and variance – their properties, Continuous probability, distributions – uniform, normal, exponential distributions, Distribution functions.

UNIT IV

Joint distribution functions, Independent random variables and their sums, conditional distributions, Joint probability distribution of functions of random variables, covariance, correlation.

UNIT V

Definition of statistics, population and sample, Representative sample, Descriptive statistics – classification and tabulation of univariate data, Graphical representation, frequency curves.

TEXTBOOKS

1. Sheldon Ross, A First course in probability (Ninth edition).

REFERENCES

1. Michael Baron, Probability and Statistics for computer scientistst.

SEMESTER-II

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
ENG 111	Basic Electronics	ES	3	0	0	3

UNIT I: ELECTRICAL QUANTITIES AND THEIR MEASUREMENT

Ohm's law, permanent magnet moving coil (PMMC) instrument, Ammeter and Voltmeter using PMMC, Measurement of resistance using Wheat Stone's Bridge and Kelvin's double bridge, measurement of capacitance using Schering's bridge and De Sautee's bridge, and measurement of inductance using Maxwell's bridge and Hay's bridge. Operation of the oscilloscope.

UNIT II: SEMICONDUCTOR DEVICES

Forward and reverse bias characteristics of PN junction diode. Design of half-wave, full wave, bridge rectifiers, clipping and clamping using PN junction diode. Bipolar junction transistors (BJTs), common-base, common-collector and common-emitter configurations using BJTs. Voltage and current gain, transistor as amplifier and buffer. Photodiode and phototransistor.

UNIT III: A.C. CIRCUITS AND OPERATIONAL AMPLIFIER

Phasor analysis, impedance and reactance, resonance, tuned circuits using R-L-C components, series reactance and resistance, parallel reactance and resistance. Characteristics of an operational amplifier, inverting and non-inverting op-amps, integrator and differentiator design using op-amp. Differential operational amplifier and common mode rejection ratio.

UNIT IV: ELECTRONIC FILTERS

Low and high frequency noise in electronic circuits, basic low-pass, high-pass, band-pass and band-reject passive filters design using resistor, capacitor and inductor. Fourier transform, magnitude and phase response, bandwidth, bode plots. Design and analysis of higher order filters. Active filter design using operational amplifier, applications of electronic filters.

UNIT V: DIGITAL LOGIC FUNDAMENTALS

Number systems: binary, decimal, octal and hexadecimal number systems, number system conversions. Logic gates: AND, OR, NOT, NAND, NOR, X-OR, X-NOR. Logic gates design using PN diodes. De Morgan's laws, Karnaugh maps. Basic combinational logic blocks: half adder, half subtractor, full adder, full subtractor, multiplexer and de multiplexer.

TEXTBOOKS

1. Principles of electronics by V K Mehta & Rohit Mehta, 2010 edition, S Chand and Co. Publisher, ISBN: 9788121924504.
2. Electronic devices and circuits by David A. Bell, 2008 edition, Oxford University Press, ISBN: 9780195693409.
3. Introduction to digital logic design by John P. Hayes, 1993 edition, Pearson Edition, ISBN: 9780201154610.

REFERENCES

1. Electronic measurements and Instrumentation by A K Sawhney, 2015 edition, Dhanpat Rai and Co., ISBN: 9788177001006.
2. Pulse, Digital and Switching waveforms by Mill man and Taube, 2011 edition, Tata McGraw Hill, ISBN: 9780071072724.

SEMESTER-II

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
ENG 111 L	Basic Electronics Lab	ES	0	0	2	1

LIST OF PRACTICAL EXPERIMENTS

1. Verification of Kirchhoff's laws (KCL, KVL).
2. Study of I-V characteristics of PN junction diode.
3. Design of half-wave rectifier using PN junction diode with and without capacitor filter.
4. Design of positive and negative clipping circuits using PN junction diodes.
5. Study of current and voltage gain characteristics of a NPN transistor in common-emitter configuration.
6. Drain characteristics of common source JFET.
7. Design of inverting and non-inverting amplifier circuits using op-amp IC 741.
8. Study of integrator and differentiator circuits using op-amp IC 741.
9. Design of Schmitt Trigger Using IC 741.
10. Study of function of digital logic gates (AND, NOT, OR, NAND, NOR).

SEMESTER-II

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 107	Data Structures	ES	3	0	0	3

UNIT I

Introduction to data structures, Abstract Data Type (ADT), representation and implementation, time and space requirements of algorithms. Array ADT, representing polynomials, sparse matrices using arrays and their operations Stacks and Queues**:** Representation and application, implementation of stack and queue operations using C.

UNIT II

Linked lists: Single linked lists, implementation of link list and various operation using C, double linked list, circular list and applications.

UNIT III: TREES

Tree terminology, binary tree, binary search tree, infix to postfix conversion, postfix expression evaluation. AVL Tree, complete binary tree representation.

UNIT IV: GRAPHS

Graph terminology, representation of graphs, path matrix, BFS (breadth first search), DFS (depth first search), topological sorting, shortest path algorithms. Priority Queues: Heap structures, binomial heaps, leftist heaps.

UNIT V: SORTING AND SEARCHING TECHNIQUES

Bubble sort, selection sort, insertion sort, quick sort, merge sort, heap sort, radix sort and implementation. Linear and binary search methods, implementation; Hashing techniques and hash functions.

TEXTBOOKS

1. "Data Structure -- A Pseudo code approach with C" by Richard R. Gilberg & Behrouz A. Forouzan, 2nd edition, 2011. Cengage Learning. Imprint: Thomson Press (India) Ltd.
2. "Data Structures Using C" by Aaron M. Tanenbaum, Yedidvah Langsam, and Moshe J. Augenstein. Pearson Publishers, 2019.

REFERENCES

1. Programming with C, Byron Gottfried, McGraw hill Education, Fourteenth reprint, 2016.
2. "Fundamental of Data Structures", (Schaums Series) Tata-McGraw-Hill.
3. Data structures and Algorithm Analysis in C, Mark Allen Weiss, Pearson publications, Second Edition Programming in C. P. Dey and M Ghosh, Second Edition, Oxford University Press.
4. "Fundamentals of data structure in C" by Horowitz, Sahani & Anderson Freed, Computer Science Press.

5. G. A. V. Pai: “Data Structures & Algorithms; Concepts, Techniques & Algorithms” Tata McGraw Hill.

SEMESTER-II

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 107 L	Data Structures Lab	ES	0	0	2	1

LIST OF PRACTICAL EXPERIMENTS

1. **Week 1 & 2: Simulate the following operations:**
   - **a.** Conversion of infix expression to postfix expression
   - **b.** Evaluation of expressions
   - **c.** **Assignment-1:** Tower of Hanoi is a mathematical puzzle where we have three rods and n disks. The objective of the puzzle is to move the entire stack to another rod, obeying the following simple rules:
     - **i.** Only one disk can be moved at a time.
     - **ii.** Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack i.e. a disk can only be moved if it is the uppermost disk on a stack.
     - **iii.** No disk may be placed on top of a smaller disk
     - **iv.** You can choose to use the function ***move (4, 1, 3, 2)***, *w*here 4 represents the number of disks. 1 represents disks on source shaft, 3 represents the destination shaft which holds the disks after the move and finally 2 represents the intermediate support shaft – temporary storage. Write a C program to simulate the given problem and: Perform the algorithmic complexity analysis for the solution you propose.
       Resources: https://www.youtube.com/watch?v=YstLjLCGmgg

1. **Week 3 & 4: Simulate the following tasks:**
   - **a.** Implementation the following operations: enqueue, dequeue and finding an element:
     - **i.** Linear Queue using arrays
     - **ii.** Circular queue arrays
     - **iii.** Priority queue singly linked list.
   - b. **Assignment-2:** The "*4-Queens Problem*" consists of placing four queens on a 4 x 4 chessboard so that no two queens can capture each other. That is, no two queens are allowed to be placed on the same row, the same column or the same diagonal (both primary and secondary diagonals). Write a C program to simulate the given problem and perform the algorithmic complexity analysis for the solution you propose.
     Reference(s): Data Structures and Program Design in C by Robert Kruse, C L Tondo, Bruce Leung and Shashi Mogalla. For pseudocode, refer the following pages 98 to 105.
     Online Reference: https://www.youtube.com/watch?v=xFv_Hl4B83A
1. **Week 5 &6: Demonstrate the following though simulation:**
   - a. Create a singly linked list and perform the following operations:
     - i. Add an element at the end of the list
     - ii. Delete an element from the beginning of the list
     - iii. Find the middle element of the list
     - iv. Search the given key form the list
     - v. Polynomial addition using linked list
     - vi. Sparse matrix operations using linked list
   - b. **Assignment-3:** Let us consider a small but busy airport with only one run-way (shown in figure). In each time unit, one plane can land or one plane can take off, but not both. Planes arrive ready to land or to take off at random times, so at any given unit of time, the runway

may be idle or a plan may be landing or taking off, and there may be several planes waiting either to land or take off. We therefore need two queues, called *landing* and *takeoff*, to hold these planes. It is better to keep a plane waiting on the ground than in the air, so a small airport allows a plane to take off only if there are no planes waiting to land. Hence, after receiving requests from new planes to land or take off, our simulation will first service the head of the queue of planes waiting to land, and only if the landing queue is empty will it allow a plane to take off. We shall wish to run the simulation through many units of time, and therefore, we embed the main action of the program in a loop that runs for cur-time (denoting current time) from 1 to a variable end-time.

Simulate the given scenario using and write the output for different inputs.

Reference(s): Data Structures and Program Design in C by Robert Kruse, C L Tondo, Bruce Leung and Shashi Mogalla. For pseudocode, refer the following pages 139 to 150.

1. **Week 7 & 8: Write code to perform the following operations:**
   a. Develop a code to test whether the given tree is binary tree or not.
   b. Implementation of Binary tree traversals techniques – pre-order, in-order, and post-order.
   c. Implementation of AVL tree and its operations
   d. Assignment-4: Given a mathematical expression, evaluate it using appropriate tree structure.

1. **Week 9 & 10: Write the codes to perform the following tasks:**
   a. Implementation of Graph traversals techniques: i) BFS and ii) DFS.
   b. Assignment-5: The **Dijkstra's algorithm** is an algorithm that gives the shortest path between two given vertices of a graph. In this problem we are given a directed graph with each edge having a non-negative weight. Thus, a solution requires a path of many other that costs least. We can think of the problem as like this: think graph G as a map of the airline routes, each node of the graph as the cities and the weights on each edge as the cost of flying from one city to another city. The solution we have to find a routing from a city v to city w such that the total cost is minimum. Write a C program to simulate the given problem. That is find the shortest path between node A and node F in the given graph. **Resource:** Data Structures and Program Design in C by Robert Kruse, C L Tondo, Bruce Leung and Shashi Mogalla. For pseudocode, refer the following pages 510 to 514.

1. **Week 11 & 12:** Implementation of the following algorithms:
   a. Linear search
   b. Binary search
   c. Implementation of Bubble sort algorithm
   d. Implementation of Selection sort algorithm
   e. Implementations of Merge sort algorithm

1. **Week 13 & 14:**
   a. Implementation of Insertion sort algorithm
   b. Implementation of quick sort algorithm
   c. Assignment-6: Suppose you work at college library. You are in the middle of a quiet afternoon when suddenly a shipment of 3928 different books arrives. The books have been dropped of in one long straight line, but they are all out of order, and the automatic sorting system is broken. To make matter worse, classes will start tomorrow, which means that first thing in the morning, students will show up in droves looking for these books. How can you get them all sorted in time?

      Simulate the given scenario using C code. Perform the algorithmic time complexity analysis for the solution you propose. Also give the space complexity.

Reference(s): Data Structures and Program Design in C by Robert Kruse, C. L. Tondo , Bruce Leung and Shashi Mogalla. For pseudocode, refer the following pages 302 to 312.
Online resources: Use the following link to get a better understanding on the problem.
https://www.youtube.com/watch?v=PgBzjlCcFvc
https://www.programiz.com/dsa/quick-sort

1. **Week 15:** Our Text editor will allow us to read a file into memory i.e., it is stored in the buffer. We consider each line of text to be a string and buffer will be a list of these lines. we shall then devise editing commands that will do list operations on lines in buffer and will do string operations on characters in a single line. Here are few commands;
    a. R – Read the text file
    b. W – Write to text file
    c. I – Insert a new line
    d. D – Delete the current line
    e. P – Previous line (back up one line in buffer)
    f. B – Go to first line of buffer
    g. E – Go to last line of buffer
    h. Q – Quit the editor

    Tasks we do are :

    a. Receiving a command from user
    b. GetCommand() – this function gets the command from user
    c. DoCommand() – this function performs the command

    Now we have to perform the command for example if the command is ‘b’ we have to go beginning of buffer; if it is ‘n’ we must move to next line. All these commands can be performed using switch case statement. Using the switch case statements we check for the command and specify the functions to perform the appropriate task.

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CHE 101	Principles of Chemistry	BS	2	0	0	2

UNIT I: CHEMICAL BONDING

Ionic, covalent, and metallic bonds. Theories of bonding: Valence bond theory, nature of covalent bond, sigma (σ) bond, Pi(π) bond. Hybridization: Types of hybridizations, sp^2, sp^3, sp^3d, d^2sp^3. Shapes of molecules (VSEPR Theory): $BeCl_2$, CO_2, BF_3, H_2O, NH_3, CH_4, PCl_5, XeF_2, SF_6, XeF_4.

Molecular orbital theory: Linear combination of atomic orbitals (LCAO Method), bond order, homo (H_2, O_2, N_2) and hetero nuclear diatomic molecules (NO, CO). Non-covalent interactions: Vander Waals interactions, dipole-ipole interactions, and hydrogen bonding.

UNIT II: PHASE RULE AND KINETICS

Phase rule: Introduction, Definition of the terms used in phase rule with examples. Application of phase rule to water system, Sulphur system and lead-silver system. Kinetics: Order and molecularity of reactions, zero order, first order and second order reactions.

UNIT III: WATER TECHNOLOGY

Standards for drinking water, Methods of Treatment of water for domestic and industrial purposes: Sedimentation, Coagulation, Filtration, Sterilization, Break point chlorination. Determination of Hardness of water by EDTA method. Demineralization of water. Softening of water: Lime-soda Process, Ion exchange process, Zeolite process. Boiler Troubles: Priming, Foaming, Scale, Sludge, Corrosion, Caustic Embrittlement.

UNIT IV: POLYMER CHEMISTRY

Classification of polymers: Natural and synthetic. Thermo plastic and Thermo setting. Degree of polymerization. Types and mechanism of polymerization: Addition (Free Radical, cationic, and anionic); condensation and copolymerization. Properties of polymers: Tg, Tactility, Molecular weight, weight average, number average and poly dispersity index. Techniques of polymerization: Bulk, emulsion, solution, and suspension.

UNIT V: ELECTOCHEMISTRY

Arrhenius theory of electrolytic dissociation, classification of electrolytes; degree of Dissociation of acids, dissociation constant of weak acids, concept of PhandpOH, buffer solutions, solubility product, common ion effect, indicators and theory of acid base indicators, conductance of solutions-specific, molar and equivalent conductance, Variation of molar conductance with dilution for strong and weak electrolytes; Migration of ions-Kohlrausch's law of independent migration of ions, Ostwald's dilution law; Nernste equation for single electrode and electrochemical cells.

TEXTBOOKS

1. Bahl and B. S. Bahl, G. D. Tuli, Essentials of physical chemistry, S Chand Publication,2014, ISBN: 8121929784.P.W. Atkins, T.L. Overton, J.P. Rourke, M.T. Weller and F.A. Armstrong Shriver and Atkins' Inorganic Chemistry, 5th Ed., Oxford University Press, London, 2010, ISBN 978-1-42-921820-7.
2. Atkins, P.W.; de Paula, J. Physical chemistry, 8th ed., 2006 Oxford University Press.

ISBN0-19-870072-5.
3. B. R. Puri, L. R. Sharma & M. S. Pathania, Principles of Physical Chemistry, 46th Ed.,2013, Vishal Publication Company.
4. F.W. Bill meyer, Textbook of Polymer Science, 3rd Ed., John Wiley & Sons, New York, 2003.

REFERENCES

1. J. Bard and L.R. Faulkner, Electrochemical methods –Fundamentals and Applications, 2nd Ed., John Wiley and Sons, 2001.
2. Jain P.C. & Monika Jain, Engineering Chemistry, Dhanpat Roy & Sons, 2015

SEMESTER-II / SEMESTER-I

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CHE 101 L	Principles of Chemistry Lab	C	0	0	2	1

LIST OF PRACTICAL EXPERIMENTS

1. Volumetric titration of HCl vs NaOH.
2. Conductometric titration of HCl vs NaOH.
3. Standardization of potassium permanganate by Oxalic acid.
4. Iodometric Determination of Ascorbic Acid (Vitamin C).
5. Determination of hardness of water by EDTA method.
6. Determination of strength of given hydrochloric acid using pH meter.
7. Estimation of iron content of the given solution using potentiometer.
8. Determination of sodium and potassium by flame photometry.

SEMESTER-II / SEMESTER-I

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
ENV 111	Environmental Science	BS	2	0	0	2

UNIT I: ENVIRONMENTAL CRISIS AND SUSTAINABLE DEVELOPMENT

Environment: Structure and functions in an ecosystem; Ecological succession; Ecological pyramids; Biosphere; Ecological systems and cycles – carbon cycle, water cycle, phosphorous cycle, nitrogen cycle, oxygen cycle; Broad nature of chemical composition of plants and animals; Natural resources covering renewable and non-renewable resources, forests, water, minerals, food and land; Energy sources, growing energy demands.

UNIT II: ECOSYSTEMS

Environmental Pollution: Structure and composition of atmosphere. Pollution – air, water, soil, thermal and radiation. Effects – acid rain, ozone layer depletion and greenhouse gas emission. Control measures. Determination of water and air quality – BOD, COD, TDS, AQI.

UNIT III: RENEWABLE AND NON-RENEWABLE RESOURCES

Environmental Biotechnology: Environmental microbiology; Biomarkers; Biosensors; Biofuels; Biotransformation; Bioremediation, factors affecting bioremediation; Molecular Ecology.

UNIT IV: BIODIVERSITY

Biodiversity and its conservation: Biodiversity hotspots; Values of biodiversity: consumptive use, productive use, social, ethical, aesthetic and option values; threats to biodiversity – habitat loss, poaching of wildlife; in-situ and ex-situ conservation.

UNIT V: POLLUTION AND POLICIES

Environmental protection and sustainability: Problems related to urban living, waste management, climate change, sustainable solutions, environmental regulation, and environmental protection acts in India and environmental ethics

TEXTBOOKS

1. Basu. M, Xavier. S. “Fundamentals of Environmental Studies”, 1st edition, Cambridge University Press, 2016.
2. Raina. M. Maier, Ian L. Pepper, Charles. P. “Environmental Microbiology” 2nd edition, Academic Press, 2004.

REFERENCES

1. Danial. D. C. “Environmental Science”, 8th edition, Jones and Barlett Publishers, MA, 2010.

SEMESTER-II / SEMESTER-I

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
ENV 111 L	Environmental Science Lab	BS	0	0	2	1

LIST OF PRACTICAL EXPERIMENTS

1. Water parameters- Test for alkalinity and turbidity of water.
2. Determination of dissolved oxygen in water.
3. Test for total suspended solids and total dissolved solids.
4. Determination of total hardness of water by EDTA titration.
5. Determination of biological oxygen demand of wastewater.
6. Determination of chemical oxygen demand of wastewater.
7. Test for iron content in river water.

SEMESTER-II

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
ISES 102	Industry Specific Employability Skills-II	HS	1	1	0	1

UNIT I: QUANTS

Average, Alligation or Mixture, Alligation or Mixture, Percentage, Profit and Loss, True discount, Partnership, Height and distance.

UNIT II: REASONING

Logical deductions, Syllogism, Image based problems, Coding and Decoding, Cubes and Cuboids, Inequalities, Input output tracing.

UNIT III: VERBAL

Ordering of sentences, Comprehension, Verbal Analogies, Essential parts of a sentence, One-word substitutes, Cause and effect, Syllogism.

UNIT IV: COMMUNICATION SKILLS

Sentence formation (Practical), Word group categorization, Casual conversation (Practical), Formal conversation (interpersonal)

TEXTBOOKS/REFERENCES

1. Mitchell S. Green – 2017, Know Thyself: The Value and Limits of Self-Knowledge.
2. Debbie Hindle, Marta Vaciago Smith - 2013 , Personality Development: A Psychoanalytic Perspective.
3. Lani Arredondo - 2000, Communicating Effectively.
4. Patsy McCarthy, Caroline Hatcher - 2002, Presentation Skills: The Essential Guide for Students.
5. Martha Davis, Elizabeth Robbins Eshelman, Matthew McKay - 2008, Time Management and Goal Setting: The Relaxation and Stress.
6. Arun Sharma – How to prepare for Quantitative Aptitude, Tata Mcgraw Hill.
7. RsAgarwal,A Modern Approach to Verbal and Non Verbal Reasoning,S.Chand Publications.
8. Verbal Ability and Reading comprehension-Sharma and Upadhyay.
9. Charles Harrington Elstor, Verbal Advantage: Ten Easy Steps to a Powerful Vocabulary, Large Print, September 2000.
10. GRE Word List 3861 – GRE Words for High Verbal Score, 2016 Edition.
11. The Official Guide to the GRE-General Revised Test, 2nd Edition, Mc Graw Hill Publication
12. English grammer and composition – S.C. Gupta.
13. R.S. Agarwal – Reasoning.
14. Reasoning for competitive exams – Agarwal.

S

SEMESTER-III

EMESTER-III

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
MAT 141	Discrete Mathematics	C	3	0	0	3

UNIT I: THE FOUNDATIONS: LOGIC AND PROOFS
Propositional Logic, Applications of Propositional Logic, Propositional Equivalences, Predicates and Quantifiers, Nested Quantifiers, Rules of Inference, Introduction to Proofs, Methods and Strategy.

UNIT II: SET THEORY
Laws of set theory, Set Operations, Functions, Sequences and Summations, Matrices

UNIT III: ELEMENTARY NUMBER THEORY, INDUCTION AND RECURSION
Divisibility and Modular Arithmetic, Integer Representations and Algorithms, Primes and Greatest Common Divisors, Solving Congruences, Mathematical Induction, Strong Induction and Well-Ordering, Recursive Definitions and Structural Induction.

UNIT IV: COUNTING PRINCIPLES
Recursive Definitions and Structural Induction, Binomial Coefficients and Identities, Applications of Recurrence Relations, Solving Linear Recurrence Relations, Divide-and-Conquer Algorithms, Recurrence Relations

UNIT V: INTRODUCTION TO GRAPH THEORY
Graphs and Graph Models, Graph Terminology and Special Types of Graphs, Trees, Spanning trees, Minimal spanning trees, Representing Graphs and Graph Isomorphism, Connectivity, Euler and Hamilton Paths, Shortest-Path Problems

TEXTBOOKS

1. Kenneth H. Rosen, Discrete Mathematics and Applications, Seventh edition, Tata McGraw-Hill, 2012.
2. J. P. Tremblay and R. P. Manohar, Discrete Mathematics with Applications to Computer Science, Tata McGraw-Hill, 1997.

REFERENCES

1. S. Lipschutz and M. L. Lipson, Schaum's Outline of Theory and Problems of Discrete Mathematics, 3rd Ed., Tata McGraw-Hill, 1999.
2. M. K. Venkataraman, N. Sridharan, and N. Chandrasekaran, Discrete Mathematics, National Publishing Company, 2003.

SEMESTER-III

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 206	Object Oriented Programming using C++	C	3	0	0	3

UNIT I: INTRODUCTION

What is object-oriented programming? Comparison of procedural programming and Object-Oriented Programming - Characteristics of Object-Oriented Languages - C++ Programming Basics: Basic Program Construction - Data Types, Variables, Constants - Type Conversion, Operators, Library Functions - Loops and Decisions, Structures - Functions: Simple Functions, passing arguments, Returning values, Reference Arguments. - Recursion, Inline Functions, Default Arguments - Storage Classes - Arrays, Strings, Addresses, and pointers. Dynamic Memory management. Linked lists in C++.

UNIT II: FEATURES OF OBJECT-ORIENTED PROGRAMMING

Introduction to Classes and Objects, Making sense of core object concepts (Encapsulation, Abstraction, Polymorphism, Classes, Messages Association, Interfaces). Constructors and its types, Destructors - Passing Objects as Function arguments and Returning Objects from Functions.

UNIT III: POLYMORPHISM

Concept of Polymorphism, Function overloading, examples and advantages of function overloading, pitfalls of function overloading, Operator overloading, Overloading unary operations. Overloading binary operators, pitfalls of operators overloading.

UNIT IV: INHERITANCE

Concept of inheritance. Derived class and based class. Derived class constructors, member function, inheritance in the English distance class, class hierarchies, inheritance and graphics shapes, public and private inheritance, aggregation: Classes within classes, inheritance, and program.

UNIT V: TEMPLATES AND EXCEPTIONS

Templates: Function templates, Class templates - Exceptions: Need of Exceptions, keywords, Simple and Multiple Exceptions - Re-throwing Exception and Exception Specifications, Custom Exception. Standard Template Library: Containers, Algorithms, iterators - potential problems with STL - Algorithms: find (), count (), sort (), search (), merge () - Function Objects: for each (), transform () - Sequence Containers: vectors, Lists, Dequeues - Iterators and specialized.

TEXTBOOKS

1. C++ Primer, Stanley B. Lippman, Stanley Lippman and Barbara Moo, Addison-Wesley Professional, Fifth edition, 2012.
2. C++: The complete reference, Schildt, Herbert. McGraw-Hill/Osborne, Fourth edition, 2017.

REFERENCES

1. Thinking in C++, Bruce, Eckel, Pearson, Second edition, Volume 1, 2002.
2. Object-oriented programming in C++, Robert Lafore, Course Sams Publishing, Fourth edition, 2001.
3. Lischner, Ray. STL Pocket Reference: Containers, Iterators, and Algorithms. " O'Reilly Media, Inc.", 2003.

SEMESTER-III

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 206 L	Object Oriented Programming using C++ Lab	C	0	0	2	1

LIST OF PRACTICAL EXPERIMENTS

Week-1:

1. Takes two integer operands and one operator form the user, performs the operation and then prints the result.
2. Generate all the prime numbers between 1 and n, where n is a value supplied by the user.
3. Searching an element in an array.
4. To find the factorial of a given integer.

Week-2:

1. Write a program to demonstrate the Inline functions.
1. Programs to understand different function call mechanism.
   a. call by reference b. call by value
1. Programs to understand storage specifiers

Week-3:

1. Write a Program to design a class having static member function named showcount() which has the property of displaying the number of objects created of the class.
1. Write a Program using class to process Shopping List for a Departmental Store. The list include details such as the Code No and Price of each item and perform the operations like Adding, Deleting Items to the list and Printing the Total value of a Order.

Week-4:

1. Write a Program which creates & uses array of object of a class.(for eg. implementing the list of Managers of a Company having details such as Name, Age, etc..).
1. Write a Program to find Maximum out of Two Numbers using friend function. Note: Here one number is a member of one class and the other number is member of some other class.

Week-5:

1. Write a Program to swap private data members of classes named as class_1, class_2 using friend function.
1. Write a Program to design a class complex to represent complex numbers. The complex class should use an external function (use it as a friend function) to add two complex numbers. The function should return an object of type complex representing the sum of two complex numbers.

Week-6:

1. Write a Program using copy constructor to copy data of an object to another object.
1. Write a Program to allocate memory dynamically for an object of a given class using class's constructor.

Week-7:

1. Write a Program to design a class to represent a matrix. The class should have the functionality to insert and retrieve the elements of the matrix
1. Write a program to design a class representing complex numbers and having the functionality of performing addition & multiplication of two complex numbers using operator overloading.

Week-8:

1. Write a Program to overload operators like *, <<, >> using friend function. The following overloaded operators should work for a class vector.
1. Write a program for developing a matrix class which can handle integer matrices of different dimensions. Also overload the operator for addition, multiplication & comparison of matrices.

Week-9:

1. Write a program to overload new/delete operators in a class.
1. Write a program in C++ to highlight the difference between overloaded assignment operator and copy construct.

Week-10:

1. Write a Program illustrating how the constructors are implemented and the order in which they are called when the classes are inherited. Use three classes named alpha, beta, gamma such that alpha, beta are base class and gamma is derived class inheriting alpha & beta
1. Write a Program to design a student class representing student roll no. and a test class (derived class of student) representing the scores of the student in various subjects and sports class representing the score in sports. The sports and test class should be inherited by a result class having the functionality to add the scores and display the final result for a student.

Week-11:

1. Write a program to maintain the records of person with details (Name and Age) and find the eldest among them. The program must use this pointer to return the result.
1. Write a Program to illustrate the use of pointers to objects which are related by inheritance.

Week-12:

1. Write a program illustrating the use of virtual functions in class.
1. Write a program to design a class representing the information regarding digital library (books, tape: book & tape should be separate classes having the base class as media). The class should have the functionality for adding new item, issuing, deposit etc. the program should use the runtime polymorphism.

Week-13:

1. Write a program to show conversion from string to int and vice-versa.
1. Write a program showing data conversion between objects of different classes.

Week-14:

1. Write a program showing data conversion between objects of different classes and conversion routine should reside in destination class.
1. Write a program to copy the contents of one file to another.

Week-15:

1. Write a program to implement the exception handling.
1. Write a program to maintain the elementary database of employee using file concepts.

SEMESTER-III

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 201	Design and Analysis of Algorithms	C	3	0	0	3

UNIT I: INTRODUCTION

Algorithmic thinking & motivation with examples, Reinforcing the concepts of Data Structures with examples. Complexity analysis of algorithms: big O, omega, and theta notation, Analysis of Sorting and Searching, Hash table, Recursive and non-recursive algorithms.

UNIT II: GENERAL PROBLEM SOLVING (GPS) TECHNIQUES

Divide and conquer: Merge sort, Quicksort, BST, Master method for Complexity analysis
Greedy method: Fractional Knapsack, Minimum spanning trees (Prim's & Kruskal's), Shortest paths: Dijkstra's algorithm, Huffman coding Dynamic Programming: 0/1 Knapsack, All-to-all shortest paths.

UNIT III

BFS & DFS, Backtracking: 8-Queen's problem, Knight's tour, Travelling Salesman Problem (TSP), Branch-and-bound: 16-puzzle problem, TSSP, Randomized algorithms: Playing Cards, Scheduling algorithms.

UNIT IV

Pattern matching algorithms: Brute-force, Boyer Moore, KMP algorithms. Algorithm analysis: Probabilistic Analysis, Amortized analysis, Competitive analysis.

UNIT V

Non-polynomial complexity: examples and analysis, Vertex cover, set cover, TSP, 3-SAT Approximation Algorithms: Vertex cover, TSP, Set cover.

TEXTBOOKS

1. Cormen, Leiserson, Rivest, Stein, "Introduction to Algorithms", 3rd Edition, MIT press, 2009.
2. Parag Dave & Himanshu Dave, "Design and Analysis of Algorithms", Pearson Education, 2008.

REFERENCES

1. Michel Goodrich, Roberto Tamassia, "Algorithm design-foundation, analysis & internet examples", Wiley., 2006.
2. A V Aho, J E Hopcroft, J D Ullman, "Design and Analysis of Algorithms", Addison-Wesley Publishing.
3. Algorithm Design, by J. Kleinberg and E. Tardos, Addison-Wesley, 2005.
4. Algorithms, by S. Dasgupta, C. Papadimitriou, and U. Vazirani, McGraw-Hill, 2006.

SEMESTER-III

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 201 L	Design and Analysis of Algorithms Lab	C	0	0	2	1

LIST OF PRACTICAL EXPERIMENTS

1. Any start up programs to teach C++ language.
2. Any start up programs to teach C++ language. Discuss linked lists and Hash table as a set of liked lists.
3. Programs for summation of series 1+X+X^2+X^3+…with different time complexities.
   Any other example of solving a problem with different time complexity programs.
4. Any two sorting techniques with time complexity analysis
   Converting recursive programs to non-recursive programs. Towers of Hanoi Problem example.
5. Binary Search Tree and Heapsort.
6. Fractional Knapsack problem, One to All shortest path (Dijkstra's algorithm).
7. Minimum spanning tree, Huffman Code.
8. All-to-all shortest paths, Transitive closure of a given directed graph using Warshall's algorithm.
9. Implement 0/1 Knapsack problem using Dynamic Programming.
10. Playing cards games simulation (Randomized algorithms), Real life events simulation.
11. Scheduling algorithms (CPU scheduling), Able and Baker problem.
12. Graph Traversal:
    a. Print all the nodes reachable from a given starting node in a digraph using BFS method.
    b. Check whether a given graph is connected or not using DFS method.
    c. 8 Queens problem, 16-puzzle problem.
13. Approximation algorithms: TSP, Vertex cover, SAT, Set Cover.
14. Any non-polynomial problems and solutions.
15. Simulation of Games and Scheduling problems.

SEMESTER-III

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
ECE 211	Digital Electronics	C	2	1	0	3

UNIT I: DIGITAL FUNDAMENTALS

4 and 5 variable K-maps, 1's and 2's complements. Codes – Binary. BCD, Excess 3. Gray, Alphanumeric codes. Sum of products and product of sums. Min terms and Maxterms. Quine-McCluskey method of minimization.

UNIT II: COMBINATIONAL CIRCUIT DESIGN

4-bit Adder and Subtractor. Binary Parallel Adder – Carry look ahead adder BCD Adder. Multiplexer. Demultiplexer. Magnitude Comparator. Decoder. Encoder. Priority Encoder.

UNIT III: SYNCHRONOUS SEQUENTIAL CIRCUITS

Flip flops – SR, JK, T, D, Master/Slave FF – operation and excitation tables. Triggering of FF. Analysis and design of clocked sequential circuits Design – Moore/Mealy models. State minimization. State assignment. Circuit implementation – Design of Counters. Ripple Counters-Ring Counters. Shift Registers. Universal Shift Register.

UNIT IV: ASYNCHRONOUS SEQUENTIAL CIRCUITS

Stable and Unstable states. Output specifications. Cycles and races. State reduction. Race free assignments. Hazards. Essential Hazards. Pulse mode sequential circuits. Design of Hazard free circuits.

UNIT V: MEMORY DEVICES

Classification of memories – ROM – ROM organization – PROM – EPROM – EEPROM –EAPROM. RAM – RAM organization – Write operation – Read operation –Programmable Logic Devices – Programmable Logic Array (PLA) – Programmable Array Logic (PAL) – Field Programmable Gate Arrays (FPGA) – Implementation of combinational logic circuits using ROM. PLA. PAL.

TEXTBOOKS/REFERENCES

1. M. Morris Mano, "Digital Design", 5th Edition, Pearson Education (Singapore) Pvt. Ltd., New Delhi, 2014.
2. John F. Wakerly, "Digital Design", Fourth Edition, Pearson/PHI, 2008.
3. John. M Yarbrough, "Digital Logic Applications and Design", Thomson Learning, 2006.
4. Charles H. Roth. "Fundamentals of Logic Design", 6th Edition, Thomson Learning, 2013.
5. Donald P. Leach and Albert Paul Malvino, "Digital Principles and Applications", 6th Edition, TMH, 2006.
6. Thomas L. Floyd, "Digital Fundamentals", 10th Edition, Pearson Education Inc, 2011.
7. Donald D. Givone, "Digital Principles and Design", TMH, 2003.

SEMESTER-III

Course Code	Course Name	Course Category	Credits

			L	T	P	C
ECE 211 L	Digital Electronics Lab	C	0	0	2	1

LIST OF PRACTICAL EXPERIMENTS

1. Realization of Basic Logic Gates.
2. Design of Code Converters (Binary to Gray) & (Gray to Binary).
3. Design of
    a. Half-Adder/Subtractor
    b. Full-Adder/Subtractor
    c. Multiplexers/De Multiplexers
    d. ALU Design
4. Design of Decoder and Encoder/ BCD 7SSD.
5. Design of Magnitude Comparator (2-bit).
6. Design and Verification of Flip-Flops using IC.
7. Design of Asynchronous Counter (Any Mod, Up and Down, Jhonson and Ring).
8. Design of Synchronous Counter (Any Mod, Decade counter 74ls90).
9. Design of Universal Shift Register (Serial to Parallel, Parallel to Serial.
10. Serial to Serial and Parallel to Parallel Converters).
11. Design & Verification of Memory (SRAM).
12. FSM Based Design Project.

SEMESTER-III

Course Code	Course Name	Course Category	Credits

			L	T	P	C
CSE 106 L	Hands on Using Python	C	0	0	4	2

LIST OF PRACTICAL EXPERIMENTS

Decision Making Control

1. Write a Python program to find the distance between two coordinate points (x1, y1) and (x2, y2).
2. Write a Python program to input Percentage. Calculate percentage and grade according to following:

   Percentage >= 90% : Grade A
   Percentage >= 80% : Grade B
   Percentage >= 70% : Grade C
   Percentage >= 60% : Grade D
   Percentage >= 40% : Grade E
   Percentage < 40% : Grade F
3. Write a Python program to find maximum between three numbers.
4. Write a Python program that computes the real roots of a quadratic function. Your program should begin by prompting the user for the values of a, b and c. Then it should display a message indicating the nature of real roots, along with the values of the real roots (if any).
5. Write a program to input angles of a triangle and check whether triangle is valid or not. Also, validate the angles entered by the user. (Sum of the three angles of triangle is 180^0)
6. Write a program to input basic salary of an employee and calculate its Gross salary according to following:

   Basic Salary <= 10000 : HRA = 20%, DA = 80%
   Basic Salary <= 20000 : HRA = 25%, DA = 90%
   Basic Salary > 20000 : HRA = 30%, DA = 95%

Looping Control

1. Write a Python program to print the sum of the series 1/2+1/3+1/4+ ... +1/N. Where N is natural number.

1. Write a Python program that prompts user to enter numbers. The process will repeat until user enters 0. Finally, the program prints sum of the numbers entered by the user.

1. Write a Python program to print all the numbers from 1 to 1000 that are not divisible by 2, 3, 5, 7, 11, 13, 17 and 19.

1. Write a Python program to find HCF (GCD) of two numbers.

1. Write a Python program to check whether a number is Armstrong number or not.

1. Write a Python program to swap first and last digits of a number.

1. Write a Python program for printing prime numbers up to N. (N>100).

1. Write a Python program to construct the following pattern, using a nested for loop.

```
*
*       *
*       *       *
*       *       *       *
*       *       *       *       *
*       *       *       *
*       *       *
*       *
*
```

1. Write a Python program to print following matrix.

1	0	1	0
0	1	0	1
1	0	1	0
0	1	0	1

Functions

1. Define a function to find sum of all odd numbers between 1 to n.
1. Define a function to check whether a number is palindrome or not.
1. Define a function to calculate the area of a circle using the formula.
1. Define a function to check whether number is perfect or not.
1. Define a function to print multiplication table of any number.
1. Define a function to print table of a number. Using this function display table of numbers from 1 to 10.
1. Define a recursive function to find power of a number.
1. Define a recursive function count number of digits in a number.
1. Write a recursive function to find a find $1^5 + 2^5 + \ldots\ldots\ldots.+n^5$.
1. Write a python program to find the factorial value of a number using recursion.
1. Write a python program to implement Tower of Hanoi using recursive function.
1. Write function for finding factors (n) and use factors function to check whether given number n is prime or not.
1. Write a python program for printing Fibonacci series
   a. Write recursive approach implementation
   b. Write iterative implementation

Files

1. Write a Python program to copy the content of one file to other file.
1. Write a Python program to number of words in the above txt file.
1. Write a Python program to number of characters without space in the above txt file.
1. Write a program that reads data from a file and print the no of vowels and constants in the file.
   1. Write a python program that accept file name as input from the user. Open the file and count the number of times a character appears in the file.

List, Tuples and Dictionary

   1. Write a Python program to create a list of each digit is a element in a list from a number. Example: Input: 5467, Output: [5,4,6,7]
   1. Write a Python program to form a number from a given list of digits Example: Input: [5, 4, 6, 7], Output: 5467
1. Write a Python program to find the second smallest number and second largest in a list.
   1. Write a python program to create dictionary of index is the key and corresponding prime number as value up to 100. Output: {1:2, 2:3, 3:5, 4:7, 5:11, 6:13, 7:17, 8:19 ……. and soon }
1. Write a Python program to find the smallest value and largest value in a dictionary.
   Example: Input: D1={1:200,2:3000,3:100,5:20} output: 20, 3000.
1. Write a Python script to generate and print a dictionary that contains a number (between 1 and n) in the form (x, x*x).
   Sample Dictionary (n = 5) :
   Expected Output : {1: 1, 2: 4, 3: 9, 4: 16, 5: 25}
1. Write a Python program to convert a list of characters into a string. Example: Input: [‘s’,’t’,’r’,’i’,’n’,’g’], Output: string.

1. Write a Python program to combine two dictionary adding values for common keys.
   d1 = {'a': 10, 'b': 20, 'c':30}
   d2 = {'a': 30, 'b': 20, 'd':40}
   Sample output: {'a': 40, 'b': 40, 'd': 40, 'c': 30}
1. Write a program to print index at which a particular value exists. If the value exists a multiple location in the list, then print all the indices. Also, count the number of times the value is repeated in the list.

1. Write a program to remove all duplicate elements in a list.

1. Write a program to create a list of numbers in the range 1 to 10. Then delete all the odd numbers from the list and print the final list.

String

1. Write a program that counts up the number of vowels contained in the string S. Valid vowels are: 'a', 'e', 'i', 'o', and 'u'. For example, if s = 'azcbobobegghakl', your program should print: number of vowels 5

1. Assume s is a string of lower-case characters. Write a program that prints the number of times the string 'bob' occurs in s. For example, if s = 'azcbobobegghakl', then your program should print Number of times bob occurs is 2.

1. Write a Python program that finds whether a given character is present in a string or not. In case if it is present then it prints the index at which it is present. Do not use built-in find functions to search the character.

1. Write a Python program that counts the occurrence of a character in a string. Do not use built-in function.

1. Write a python program for following:

a. Take a input string with spaces, split it into list of words

b. From the list of words, create dictionary with keys (only unique words) and values (length of the word)

1. Write a python program to count number of vowels, spaces and to find longest word in a given input string. (Take input string with spaces)

1. Write a python program to reverse a string. Do not use inbuilt function.

Searching and Sorting

1. Write a Python program for binary search algorithm.

1. Write a Python program for linear search algorithm.

1. Write a Python program to display the elements in an ascending order using bubble sort algorithm.

1. Write a Python program to display the elements in a descending order using selection sort algorithm.

Object Oriented Programming

1. Write a Python program to create a student class (id, name, mid1_marks, mid2_marks, quiz_marks). Create a student objects and write a function marksList() to display student's result as given below:

   ROLL NUMBER:
   NAME:
   MID1:
   MID2:
   QUIZ:
   TOTAL: MID1+MID2+QUIZ

RESULT: A GRADE (IF TOTAL>=80), B GRADE (TOTAL<80 and TOTAL>=60), C GRADE (TOTAL>=50 and TOTAL<60)

(Assume that maximum marks for mid_term1 and mid2_marks is 25 each , and quiz_marks is 50).

1. Write a Python program to create a EMP class (id, name, sal), create employee objects and write a function PaySlip(empobj) to display particular employee Pay Slip as given below:

EMP ID:
EMP NAME:
EMP BASIC: It is equal to sal.
EMP HRA:
EMP DA:
EMP TAX:
EMP GROSS SAL: BASIC (sal) +HRA (18% of sal) +DA (10% of sal)
EMP NET SAL: GROSS SAL-10% of GROSS SAL

1. Write a Python program to define rectangle class with field's length and breadth. Define color rectangle class which is inherited from rectangle class with additional field color. Create N color rectangle objects and print which color rectangle is having minimum area.
1. Write a Python program to define CAR class (model, speed, price) and Firing CAR class which inherits from CAR with additional field number of bullets and fire method ().
1. Write a Program in python using object-oriented concept to create a base class called Polygon and there are three derived classes named as triangle, rectangle and square.

i. The base class consists of the input function for accepting sides length
ii. The derived classes must have output function for displaying area of triangle, rectangle and square.

SEMESTER-III

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 230	Industry Standard Coding Practice -1	ES	0	0	4	1

UNIT I
Problem Solving with - Basic coding practices, Expression Evaluation, Operators Usage, Expressions, Control Structures, Loop & Iterations for all test case scenarios.

UNIT II
Problem Solving using time efficient logics, linear list data, Array problems, 2D Arrays and Matrix Data for all test case scenarios.

UNIT III
Problem Solving with - Pointers &
Memory referencing, String Handling, functions for all test case scenarios.

UNIT IV
Problem Solving with - parameter passing, Recursions, Recursion Analysis, Structures and unions, Enumerations & Memory allocation for all test case scenarios.

UNIT V
Problem solving with - String manipulations. Lists, display patterns, strings, matrix, tuples, dictionaries, modules, packages, exception handling using Python.

TEXTBOOKS

1. Problem solving with C++ -9e- Walter Savitch – Pearson.
2. The complete Reference C, Fourth R Edition – Herbert Schildt – MC Graw Hill.
3. Programming in Python 3, A complete introduction to Python language - 2e - Mark Summerfield – Addison-Wiley.

SEMESTER-III

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
ECO 121	Principles of Economics	HS	3	0	0	3

UNIT I: INTRODUCTION TO ECONOMICS

Why study economics? Scope and method of economics; the economic problem: scarcity and choice; the question of what to produce, how to produce and how to distribute output. Science of economics; the basic competitive model; prices. Science of economics; the basic competitive model; prices. Property rights and profits; incentives and information; rationing. Opportunity sets; economic systems; reading and working with graphs.

UNIT II: DEMAND AND SUPPLY

Determinants of individual demand/supply; demand/supply schedule and demand/supply curve; market versus individual demand/supply. Shifts in the demand/supply curve, demand, and supply together. How prices allocate resources, elasticity, and its application. How prices allocate resources, elasticity, and its application. Controls on prices; taxes and the costs of taxation. Consumer surplus; producer surplus and the efficiency of the markets.

UNIT III: CONSUMER THEORY

The consumption decision - budget constraint. The consumption decision - budget constraint, consumption, and income/price changes. Demand for all other goods and price changes. M\references (indifference curves); properties of indifference curves. Utility and preferences (indifference curves); properties of indifference curves. Consumer ‘s optimum choice. Income and substitution effects. Income and substitution effects. Applying consumer theory: Labour.

UNIT IV: PRODUCER THEORY

Production, short- run production function and returns to factor. Production, short- run production function and returns to factor. Production, short- run production function and returns to factor. Average-marginal relationship. Long– run production function and laws of return to scale- role of technology. Long– run production function and laws of return to scale- role of technology. Cost function and cost structure of a firm in the short- run. Long run cost function and cost structure.

UNIT V: TYPES OF MARKET

Perfect competition -features. Perfect competition- profit maximization. Shut-down and break-even points. Monopoly: marginal revenue; marginal cost; profit maximization. Shutdown rule; market power; price discrimination. Monopolistic competition and product differentiation

TEXTBOOKS

1. Principles of microeconomics, N. Gregory Mankiw, Publisher: Cengage Learning 5th edition.
2. Perloff, Jeffrey M. Microeconomics. 5th ed. Addison Wesley, 2008. ISBN: 9780321558497.

SEMESTER-III

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
ISES 201	Industry Specific Employability Skills-III	HS	1	1	0	1

UNIT I: QUANTS

Numbers, Problems on numbers (Divisibility, power cycle, reminder cycle), Problems on ages, Problems on HCF and LCM, Simple interest, compound interest, Data interpretation (Charts, tables, pie charts, lines).

UNIT II: REASONING

Direction sense, Direction sense, Logical order, Analytical reasoning, Passage and inference, Selection decision table, Attention to details, Seating arrangements

UNIT III: VERBAL

Spellings, Selecting words, Spotting errors, Ordering of words, Sentence correction, Sentence improvement, Synonyms, Antonyms.

UNIT IV: COMMUNICATION SKILLS

Topic wise discussion, Group discussion, Debate, Presentations.

TEXTBOOKS/REFERENCES

1. Mitchell S. Green – 2017, Know Thyself: The Value and Limits of Self-Knowledge.
2. Debbie Hindle, Marta Vaciago Smith - 2013 , Personality Development: A Psychoanalytic Perspective.
3. Lani Arredondo - 2000, Communicating Effectively.
4. Patsy McCarthy, Caroline Hatcher - 2002, Presentation Skills: The Essential Guide for Students.
5. Martha Davis, Elizabeth Robbins Eshelman, Matthew McKay - 2008, Time Management and Goal Setting: The Relaxation and Stress.
6. Arun Sharma – How to prepare for Quantitative Aptitude, Tata Mcgraw Hill.
7. RsAgarwal,A Modern Approach to Verbal and Non Verbal Reasoning,S.Chand Publications.
8. Verbal Ability and Reading comprehension-Sharma and Upadhyay.
9. Charles Harrington Elstor, Verbal Advantage: Ten Easy Steps to a Powerful Vocabulary, Large Print, September 2000.
10. GRE Word List 3861 – GRE Words for High Verbal Score, 2016 Edition.
11. The Official Guide to the GRE-General Revised Test, 2nd Edition, Mc Graw Hill Publication
12. English grammer and composition – S.C. Gupta.
13. R.S. Agarwal – Reasoning.
14. Reasoning for competitive exams – Agarwal.

SEMESTER-IV

SEMESTER-IV

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
MAT 131	Differential Equations	BS	3	0	0	3

UNIT I: FIRST ORDER DIFFERENTIAL EQUATIONS
Geometric meaning of y'=f(x, y), Direction Fields, Euler's Method, Classification of ODEs (Linear, Non-linear, Exact, Separable, Integrating Factor, Bernoulli Equations, Initial Value Problem, Modelling (Free falling object, Radioactivity, RL-circuit).

UNIT II: SECOND AND HIGHER ORDER LINEAR ODES
Homogeneous Linear ODEs, Modelling of Free Oscillations of a Mass-Spring System, Euler-Cauchy Equations, Non-homogeneous ODEs, Variation of Parameters, Modelling (Forced Oscillations, Electric Circuits)

UNIT III: SYSTEM OF ODES
Modelling Engineering problems (Electric Network, Mixing problem in two tanks etc.)
as systems of ODEs, Wronskian, Phase-Plane Method, Critical Points & Stability, Qualitative Methods for Nonlinear Systems, Nonhomogeneous Linear Systems of ODEs.

UNIT IV: SERIES SOLUTIONS OF ODES
Introduction to power series method, Legendre's equation & polynomials, Frobenius Method, Bessel's Equations & Functions

UNIT-V: LAPLACE TRANSFORMS
Laplace transforms of standard functions, Shifting Theorems, transforms of derivatives and integrals, Unit step function, Dirac's delta function, Inverse Laplace transforms, Convolution theorem (without proof), Application: Solutions of ordinary differential equations using Laplace transforms.

TEXTBOOKS

1. William Boyce and Richard DiPrima, Elementary Differential Equations and Boundary Value Problems, 11^{th} Edition, Wiley-India.
2. Erwin Kreyszig Advanced Engineering Mathematics, 10^{th} Edition, Wiley-India.
3. Mary L. Boas, Mathematical Methods in Physical Sciences, 3rd Edition, Wiley-India.

REFERENCES

1. Mary L. Boas, Mathematical Methods in Physical Sciences, 3rd Edition, Wiley-India.
2. S. Vaidyanathan, Advanced Applicable Engineering Mathematics, CBS Publishers.

SEMESTER-IV

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 204	Computer Organization and Architecture	C	3	0	0	3

UNIT I: BASIC STRUCTURE OF COMPUTERS

Functional units – Bus structures – Instruction set architecture: Instruction formats - addressing modes - Architecture and instruction set of 8086/8088 microprocessor- Assembly language programming - Fixed point and floating-point operations – ALU design.

UNIT II: BASIC PROCESSING UNIT

Fundamental concepts – Execution of a complete instruction – Hardwired control – Micro programmed control design- Nano programming- CISC-RISC- principles.

UNIT III: PIPELINE PROCESSING

Basic concepts, instruction and arithmetic pipeline, data hazards, control hazards and structural hazards, techniques for handling hazards. Pipeline optimization techniques: Compiler techniques for improving performance.

UNIT IV: MEMORY SYSTEM

Basic concepts – Semiconductor RAM – ROM – Speed – Size and cost – Cache memories – Improving cache performance – Virtual memory – Memory management requirements– Associative Memories-Secondary-storage-devices.

UNIT V: I/O ORGANIZATION

Accessing I/O devices – Programmed Input/output - Interrupts – Direct Memory Access– Interface circuits – Standard I/O Interfaces - I/ O devices and Processors.

TEXTBOOKS

1. Computer Organization, Carl Hamacher, Zvonko Vranesic and Safwat Zaky, V Edition, McGraw-Hill publications.
2. "Computer Organization and Architecture – Designing for Performance", William Stallings, Ninth edition, Pearson publications.

REFERENCES

1. Computer System Architecture, Morris Mano, Third edition, Pearson publications.
2. Andrew S. Tanenbaum, "Structured Computer Organization",
3. David A. Patterson and John L. Hennessy, "Computer Organization and Design: The Hardware/Software interface"
4. John P. Hayes, "Computer Architecture and Organization", Third Edition, Tata McGraw Hill.
5. An Introduction to 8086/8088 Assembly Language Programming, Thomas P. Skinner, John Wiley & Sons, 1985.

SEMESTER-IV

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 204 L	Computer Organization and Architecture Lab	C	0	0	2	1

LIST OF PRACTICAL EXPERIMENTS

1. Write Assembly language program to print the numbers from 0 to 9.
2. Write Assembly language programs to find average of numbers stored in an array.
3. Write Assembly language programs to find the largest number in an array.
4. Write Assembly language programs to sort the numbers in ascending order.
5. Write Assembly language programs to find L.C.M of two numbers.
6. Write Assembly language programs to find G.C.D of two numbers.
7. Write Assembly language programs to display nth term Fibonacci number.
8. Write Assembly language programs to find the factorial of a number.
9. Programs for 16-bit Arithmetic Operations for 8086 (Using Microprocessor trainer kit 8086).
10. Program for String Manipulations for 8086 (Using Microprocessor trainer kit 8086).
11. Develop an assembler to convert the given assembly language program into machine language program by considering 8086/88 microprocessor.
12. Develop a simulator for 8086/88 microprocessor.

SEMESTER-IV

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 301	Operating Systems	C	3	0	0	3

UNIT I: OPERATING SYSTEMS OVERVIEW

Operating system overview-objectives and functions, Evolution of Operating System- Computer System Organization- Operating System Structure and Operations- System Calls, System Programs, OS Generation and System Boot.

UNIT II: PROCESS SCHEDULING

Processes-Process Concept, Process Scheduling, Operations on Processes, Inter process Communication; CPU Scheduling algorithms; OS – examples.

UNIT III: **PROCESS SYNCHRONIZATION AND DEADLOCKS**

Threads- Overview, Multithreading Models; Process Synchronization – Critical Section Problem, Mutex Locks, Semaphores, Monitors; Deadlocks- OS examples.

UNIT IV: STORAGE MANAGEMENT

Main Memory-Contiguous Memory Allocation, Segmentation, Paging, Virtual Memory- Demand Paging, Page Replacement, Allocation, Thrashing; OS examples.

UNIT V: STORAGE MANAGEMENT

I/OSYSTEMS: Mass Storage Structure- Overview, Disk Scheduling and Management; File System Storage-File Concepts, Directory and Disk Structure, Sharing and Protection; File System Implementation- File System Structure, Directory Structure, Allocation Methods, Free Space Management- OS examples.

TEXTBOOKS

1. Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, “Operating System Concepts”, 9th Edition, John Wiley and Sons Inc.

REFERENCES

1. William Stallings, “Operating Systems – Internals and Design Principles”, 9th Edition, Pearson publications.
2. Andrew S. Tanenbaum, “Modern Operating Systems”, Fourth Edition, Pearson publications.
3. Harvey M. Deitel, Paul J. Deitel, David R. Choffnes (Author)“Operating Systems”, Third Edition.

SEMESTER-IV

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 301 L	Operating Systems Lab	C	0	0	2	1

LIST OF PRACTICAL EXPERIMENTS

Shell Programs

1. Write a script to find the greatest of three numbers (numbers passed as command line parameters).
2. Write a script to check whether the given no. is even/odd.
3. Write a script to calculate the average of n numbers.
4. Write a script to check whether the given number is prime or not.
5. Write a script to check whether the given input is a number or a string.
6. Write a script to compute no. of characters and words in each line of given file.
7. Write a script to print the Fibonacci series up to n terms.
8. Write a script to calculate the factorial of a given number.
9. Write a script to calculate the sum of digits of the given number.
10. Write a script to check whether the given string is a palindrome.
11. Write a shell script that accepts a string from the terminal and echo a suitable message if it doesn't have at least 5 characters including the other symbols.
12. Write a shell script to echo the string length of the given string as argument.
13. Write a shell script that accepts two directory names as arguments and deletes those files in the first directory which are similarly named in the second directly. Note: Contents should also match inside the files.
14. Write a shell script to display the processes running on the system for every 30 seconds, but only for 3 times.
15. Write a shell script that displays the last modification time of any file.
16. Write a shell script to check the spellings of any text document given as an argument.
17. Write a shell script to encrypt any text file.
18. Combine the above commands in a shell script so that you have a small program for extracting a wordlist.
19. Write a shell script which reads the contents in a text file and removes all the blank spaces in them and redirects the output to a file.
20. Write a shell script that changes the name of the files passed as arguments to lowercase.
21. Write a shell script to translate all the characters to lower case in a given text file.
22. Write a shell script to combine any three text files into a single file (append them in the order as they appear in the arguments) and display the word count.
23. Write a shell script that, given a file name as the argument will write the even numbered line to a file with name evenfile and odd numbered lines to a file called oddfile.
24. Write a shell script which deletes all the even numbered lines in a text file.
25. Write a script called hello which outputs the following: • your username • the time and date • who is logged on • also output a line of asterices (*********) after each section.
26. Write a script that will count the number of files in each of your subdirectories.
27. Write a shell script like a more command. It asks the user name, the name of the file on command prompt and displays only the 15 lines of the file at a time on the screen. Further, next 15 lines will be displayed only when the user presses the enter key / any other key.
28. Write a shell script that counts English language articles (a, an, the) in a given text file.

29. Write the shell script which will replace each occurrence of character c with the characters chr in a string s. It should also display the number of replacements.
30. Write a shell program to concatenate to two strings given as input and display the resultant string along with its string length. Write a shell program to simulate a simple calculator. 90) Write a shell program to count the following in a text file. • Number of vowels in a given text file. • Number of blank spaces. • Number of characters. • Number of symbols. • Number of lines

CPU scheduling algorithms

- First Come First Serve
- Shortest Job First
- Priority
- Round Robin

Semaphore and Deadlock

- Write a C program to implement the Producer & consumer Problem using Semaphore.
- Write a C program to simulate Bankers algorithm for the purpose of deadlock avoidance.

Page Replacement Algorithms

- First In First Out
- Least Recently Used
- Optimal
- Least Frequently Used
- Second Chance

SEMESTER-IV

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 207	Java Programming	C	3	0	0	3

UNIT I: INTRODUCTION TO JAVA

An Overview of Java - Data types, Variables and Arrays, operators, expressions, Control statements, Classes, Objects, Constructor, Methods, this reference, static keyword, and final keyword; String handling, Compiling using command line argument; Inheritance - Concept, Member access, Abstract Class, Interface, Creating Multilevel hierarchy- super uses, Packages-access specifiers, using final with inheritance; Polymorphism - Compile time Polymorphism, Method overloading, Constructor overloading; Run time polymorphism, Method overriding, Dynamic method dispatch.

UNIT II: EXCEPTION HANDLING & MULTITHREADING

Fundamentals of exception handling, Uncaught exceptions, using try and catch, multiple catch blocks, Exception types - Introduction to Object class, Exception class hierarchy, Termination or presumptive models, Built-in exceptions, User defined exceptions, Nested try statements, Throw, Throws, and Finally. Multithreading- Differences between thread-based multitasking and process-based multitasking, Java thread model, Thread life cycle, Creating threads – Thread class, Runnable interface, Thread priorities, Synchronizing threads, Inter-thread communication.

UNIT III: STREAM BASED I/O (JAVA.IO)

Java API, The Stream Classes-Byte streams and Character streams, reading console Input and Writing Console Output, File class, Reading and writing Files, Random access file operations, The Console class, Serialization, Enumerations, auto boxing, generics.

UNIT IV: THE COLLECTIONS FRAMEWORK (JAVA.UTIL) & JDBC

Collection's overview, Collection Interfaces, The Collection classes- Array List, Linked List, Hash Set, Tree Set, Priority Queue, Array Deque, and other utility classes. Accessing a Collection via an Iterator, using an Iterator, The For-Each alternative, Map Interfaces and Classes, Comparators, Collection algorithms, String Tokenizer. JDBC – What is database, Table, SQL Syntax-Create, Insert, Select, Drop, Alter, Update, Delete, what is JDBC, JDBC Architecture and Components, JDBC Driver Types, Connections, Statements, Result Set.

UNIT V: GUI PROGRAMMING WITH SWING

Introduction - AWT & Swings, MVC architecture, components, containers. Understanding Layout Managers, Flow Layout, Border Layout, Grid Layout, Card Layout, Grid Bag Layout. Event Handling- The Delegation event model- Events, Event sources, Event Listeners, Event classes, Handling mouse and keyboard events, Adapter classes, Inner classes, Anonymous Inner classes. A Simple Swing Application, Applets – Applets and HTML, Security Issues, Applets and Applications, passing parameters to applets. Creating a Swing Applet, painting in Swing, A Paint example, Exploring Swing Controls- J Label and Image Icon, J Text Field, The Swing Buttons- J Button, J Toggle Button, J Check Box, J Radio Button, J Tabbed Pane, J Scroll Pane, J List, J Combo Box, Swing Menus, Dialogs.

TEXTBOOKS

1. Java The complete reference, 11th edition, Herbert Schildt, McGraw Hill Education (India) Pvt. Ltd.

REFERENCES

1. Understanding Object-Oriented Programming with Java, updated edition, T. Budd, Pearson Education.
2. An Introduction to programming and OO design using Java, J. Nino and F.A. Hosch, John Wiley & sons.
3. Introduction to Java programming, Y. Daniel Liang, Pearson Education.
4. Object Oriented Programming through Java, P. Radha Krishna, and Universities Press.
5. Programming in Java, S. Malhotra, S. Chaudhary, 2nd edition, Oxford Univ. Press.
6. Java Programming and Object-Oriented Application Development, R. A. Johnson, Cengage Learning.

SEMESTER-IV

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 207 L	Java Programming Lab	C	0	0	2	1

LIST OF PRACTICAL EXPERIMENTS

1. Declare a class named Teacher. The class will have all the data members as per your convenient. The class will have constructors. Write a function to read the values of the class variables. The values of the variable will be stored in a FILE (text file). The values will be stored in a structured format of your own choice.
Further, read the content of the FILE and display the content in an ordered form (First Name, Last Name).

Concept Learning:
1. FILE manipulation
2. Use try catch blocks
3. Use multiple try catch block
4. Finally statement
Try to have your own Exception

2. Create three classes named Student, Teacher, Parents. Student and Teacher class inherits Thread class and Parent class implements Runnable interface. These three classes have run methods with statements. The task of the teacher class of the first assignment has to be synchronized. Similarly, the other two classes should have run methods with few valid statements under synchronized.

3. Create two classes named Student and Teacher with required data members. Assume that the information about the Student and Teacher is stored in a text file. Read n and m number of Student and Teacher information from the File. Store the information in Array list of type Student and Teacher Array List<Student> and Array List<Teacher>. Print the information of Teacher who taught OOPS and Maths. Use Iterator and other functions of util in your program.

4. Watch any of the favorite movie of your choice (any language is fine, preferably English). Create a Text file to store at least 10 meaningful dialogs from the movie and store it in a text file. Process the file to remove the stop words (eg. the, is, was, …….) and create another file to have clean text (word).

5. Write a java program to create Hashtable to act as a dictionary for the word collection. The dictionary meaning of the words, including synonyms, etc., has to be displayed.

6. Declare two classes Student and Teacher. The classes will have the data members and constructors as per your convenience. Write a JAVA program, (i) where the Teacher will enter the marks of the all the students in the database. (ii) Once the marks are entered, the student can view the marks.

7. Create GUI for the above program to upload the dialog FILE, clean the FILE. The GUI should take input from the user for invoking the dictionary for displaying dictionary meaning.

8. Declare a class named Teacher. The class will have all the data members as per your convenient. The class will have constructors. Develop a GUI to read the values of the class variables from the

keyboard. Use text field to read the values. Use button to store it in a file one by one. The values will be stored in a structured format of your own choice.
Have an option in the GUI to search the name of the students by roll number and display the content in the test field.

9. Create two classes named Student and Teacher with required data members. Read the information about the student and teacher using text fields. Use checkbox to choose the option to feed either teacher information or student information. Store the information about the Student and Teacher in a text file. Read n and m number of Student and Teacher information from the File. Show in the GUI about a Teacher who taught two subjects to a section. Develop at least one of the applications (AWT problem) using swing package.

10. Create a Window based applications using various controls to handle subject registration for exams. Have a List Box to display the subject of semesters. Have one more List box having subject codes. Have a combo box to select the Semester, which will change the list of course and code in the list boxes. Display the subject registered for the examination on the right side of the window.

11. Declare a class named Teacher. The class will have all the data members as per your convenient. The class will have constructors. Develop a GUI to read the values of the class variables from the keyboard. Use text field to read the values. Use button to store it in a file one by one. The values will be stored in a structured format of your own choice.
Have an option in the GUI to search the name of the students by roll number and display the content in the test field. Develop at least one of the applications (AWT problem) using swing package.

12. Create a Window based application for displaying your photo album. Create a Frame and Canvas. Change the border, foreground and background colors of canvas and other controls. Have buttons to start the image show, pause the image show and end the image show. Explore the options to play background music.

13. Create a Window application with menu bar and menu. The frame will also have a text area with scroll bar. In the menu, have File related options. Open a file and its content has to be displayed in the text area.

14. Create a GUI using various controls: (i) to upload the marks of all the students presented in a marks.csv or marks.txt file into the database. (ii) to show the marks of the respective student after uploading the marks into the database. Note: Handle the exception, if the file is not present (or) if the marks are not uploaded in the database.

15. Individual Project. Every student should do a project to achieve all the course outcomes. Based on the course outcomes, the project will be evaluated.

SEMSTER-IV

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 203	Formal Languages and Automata Theory	C	3	0	0	3

UNIT I: FUNDAMENTALS

Strings, Alphabet, Language, Operations, Finite state machine, definitions, finite automaton model, acceptance of strings, and languages, deterministic finite automaton and non-deterministic finite automaton, transition diagrams and Language recognizers.

Finite Automata: NFA with Î transitions - Significance, acceptance of languages. Conversions and Equivalence: Equivalence between NFA with and without Î transitions, NFA to DFA conversion, minimisation of FSM, equivalence between two FSM's, Finite Automata with output- Moore and Melay machines.

UNIT II: REGULAR LANGUAGES

Regular sets, regular expressions, identity rules, constructing finite Automata for a given regular expressions, Conversion of Finite Automata to Regular expressions. Pumping lemma of regular sets, closure properties of regular sets (proofs not required).

Grammar Formalism: Regular grammars-right linear and left linear grammars, equivalence between regular linear grammar and FA, inter conversion, Context free grammar, derivation trees, sentential forms. Right most and leftmost derivation of strings.

UNIT III: CONTEXT FREE GRAMMARS

Ambiguity in context free grammars. Minimisation of Context Free Grammars. Chomsky normal form, Greiback normal form, Pumping Lemma for Context Free Languages. Enumeration of properties of CFL (proofs omitted). Push Down Automata: Push down automata, definition, model, acceptance of CFL, Acceptance by final state and acceptance by empty state and its equivalence. Equivalence of CFL and PDA, interconversion. (Proofs not required). Introduction to DCFL and DPDA.

UNIT IV: TURING MACHINE

Turing Machine, definition, model, design of TM, Computable functions, recursively enumerable languages. Church's hypothesis, counter machine, types of Turing machines (proofs not required). linear bounded automata and context sensitive language.

UNIT V: COMPUTABILITY THEORY

Chomsky hierarchy of languages, linear bounded automata and context sensitive language, LR(0) grammar, decidability of, problems, Universal Turing Machine, undecidability of posts. Correspondence problem, Turing reducibility, Definition of P and NP problems, NP complete and NP hard problems.

TEXTBOOKS

1. "Introduction to Automata Theory Languages and Computation". Hopcroft H.E. and Ullman J. D. Pearson Education
2. Introduction to Theory of Computation – Sipser 2nd edition Thomson

REFERENCES

1. Introduction to Forml languages Automata Theory and Computation Kamala Krithivasan Rama R.
2. Introduction to Computer Theory, Daniel I.A. Cohen, John Wiley.
3. Theory Of Computation: A Problem - Solving Approach, Kavi Mahesh, Wiley India Pvt. Ltd.

SEMESTER-IV

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
ISES 202	Industry Specific Employability Skills-IV	HS	1	1	0	1

UNIT I: QUANTS

Logarithms. Permutations and combinations. Probability. Progressions, Geometry and Mensuration, Geometry and Mensuration

UNIT II: REASONING

Statement and conclusions, Most logical choice, Inferred meaning, Data arrangements, Venn diagram, Flow charts and logical gates, Puzzles, Case lets, Ordering, Ranking, Grouping.

UNIT III: VERBAL

Classification of sentences, Logical sequence of words, Verbal reasoning Analyzing arguments, Verification of truth, Matching definitions, Theme detection, Idioms and phrases Antonyms Synonyms

UNIT IV: COMMUNICATION SKILLS

Conditionals, Tense Forms, Verb Forms.

UNIT V: VERBAL ABILITY

Extempore, JAM, Active listening, Email Etiquette, Self-image and self-presentation, FAQ's, Resume

TEXTBOOKS/REFERENCES

1. Mitchell S. Green – 2017, Know Thyself: The Value and Limits of Self-Knowledge.
2. Debbie Hindle, Marta Vaciago Smith - 2013 , Personality Development: A Psychoanalytic Perspective.
3. Lani Arredondo - 2000, Communicating Effectively.
4. Patsy McCarthy, Caroline Hatcher - 2002, Presentation Skills: The Essential Guide for Students.
5. Martha Davis, Elizabeth Robbins Eshelman, Matthew McKay - 2008, Time Management and Goal Setting: The Relaxation and Stress.
6. Arun Sharma – How to prepare for Quantitative Aptitude, Tata Mcgraw Hill.
7. RsAgarwal,A Modern Approach to Verbal and Non Verbal Reasoning,S.Chand Publications.
8. Verbal Ability and Reading comprehension-Sharma and Upadhyay.
9. Charles Harrington Elstor, Verbal Advantage: Ten Easy Steps to a Powerful Vocabulary, Large Print, September 2000.
10. GRE Word List 3861 – GRE Words for High Verbal Score, 2016 Edition.
11. The Official Guide to the GRE-General Revised Test, 2nd Edition, Mc Graw Hill Publication
12. English grammer and composition – S.C. Gupta.
13. R.S. Agarwal – Reasoning.
14. Reasoning for competitive exams – Agarwal.

SEMESTER-IV

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 330	Industry Standard Coding Practice -2	ES	0	0	4	1

UNIT I

Problems Solving with: Structure Pointers, formation of links, Operations on Linked lists, Operations on a circular linked list, Operations on a double linked list & Industry Standard Practice Questions.

UNIT II

Problem Solving with - Stack Operations, Queue data structure Implementation, Linear / Binary Search Algorithms, Sorting Algorithms, Industry Standard Practice Questions.

UNIT III

Problem Solving with - Nonlinear data structures, trees operations, application of search property on a binary tree, tree balancing.

UNIT IV

Problem Solving with - Multiway search structures, Operations on a 2-4 tree, nonlinear structures, red, black trees & operations, Tries, String Algorithms & Industry Standard Practice Questions.

UNIT V

Problem Solving with – features of Object-oriented programming, leveraging Standard Template Libraries. Industry Standards of leveraging DBMS concepts, SQL Queries, Entity Relationship Models, Query Optimization, Transactions & Concurrency, Normalization & Industry Standard Practice Questions.

TEXTBOOKS

1. Fundamentals of Data Structures in C++ - 2e- Sahni Horowitz - Universities Press.
2. Algorithms -4e- Robert Sedgewick & Kevin Wayne - Addison-Wesley Professional.
3. C++ Standard Library A Tutorial and Reference – 2e - Nicolai M. Josuttis - Addison Wesley Longman.
4. An Introduction to Database Systems – 8e - C.J. Date – Pearson.
5. Competitive Programming – 3e – Steven Halim, Felix Halim

SEMESTER-V

SEMESTER-V

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
MAT 211	Linear Algebra	BS	3	0	0	3

UNIT I: VECTOR SPACE

Elimination, LU factorization, null-spaces and other subspaces, bases and dimensions, vector spaces, complexity.

UNIT II: FACTORIZATION

Orthogonality, projections, least-squares, QR, Gram–Schmidt, orthogonal functions.

UNIT III: MATRICES

Eigenvectors, determinants, similar matrices, Markov matrices, ODEs, symmetric matrices, definite matrices.

UNIT IV: ITERATIVE METHODS

Defective matrices, SVD and principal-components analysis, sparse matrices and iterative methods, complex matrices, symmetric linear operators on functions.

UNIT V: APPLICATIONS

Matrices from graphs and engineering.

TEXTBOOKS

1. G. Strang, Linear Algebra and Its applications, Nelson Engineering, 4th Edn., 2007.
2. K. Hoffman and R. Kunze, Linear Algebra, Prentice Hall of India, 1996.

REFERENCES

1. S. Axler, Linear Algebra Done Right, 2nd Edn., UTM, Springer, Indian edition, 2010.
2. G. Schay, Introduction to Linear Algebra, Narosa, 1997.

SEMESTER-V

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 303	Computer Networks	C	3	0	0	3

UNIT I: OVERVIEW OF THE INTERNET (PHYSICAL LAYER AND DATA LINK LAYER)
Basic Computer Network concepts, Protocol, Layering Scenario. Layer Architecture: OSI Model, TCP/IP model. Internet history standards and administration; Comparison of the OSI and TCP/IP reference model. Guided transmission media, wireless transmission media. Different LAN topologies: BUS, RING and STAR topology. Data Link layer design issues: Error detection techniques. Error Correction Techniques, Flow control. Sliding Window protocols. Go back N and selective Repeat protocols. Difference between single bit sliding window and n-bit sliding window protocols.

UNIT II: MEDIUM ACCESS CONTROL
Static and Dynamic channel Allocations. Shared channel Access: Pure ALOHA and slotted ALOHA. Persistent CSMA protocols: 1, P and Non-persistent CSMA protocols. CSMA with collision detection. Comparison of different CSMA protocols. Collision free protocols: Bit-map protocol, Token Ring and Binary Count down protocols. Limited Contention protocols: Adaptive tree walk protocol. Shared medium for wireless networks: CSMA/CA or MACA. Interconnecting LANs: HUBS, Repeaters and Switches and bridges. Spanning tree algorithm for bridges.

UNIT III: NETWORK LAYER
Overview: Connection oriented and connection less services. Comparison of packet switched, and circuit switched networks. Routing: proactive routing and reactive routing protocols, static and dynamic routing protocols. Dijkstra Algorithm, Distance vector routing and Link state routing protocols. Routing in wireless networks: AODV and DSR routing protocols. Overview of IP header and IP addressing. Classful IP addressing: Class A, B, C, D and E. Limitations of classful Addressing, Introduction to Subnet. Overview of Congestion: Warning Bit, Choke packets, Load Shedding, RED (Random Early Detection).

UNIT IV: INTERNETWORKING AND TRANSPORT LAYER
IP Encapsulation and Tunneling. IP packet fragmentation, ICMP, ARP. ICMP, DHCP, Introduction to Transport layer. Different end-to-end transport layer protocols: TCP and UDP. Brief explanation of TCP protocol. Packet formats for TCP and UDP protocol.

UNIT V: TRANSPORT AND APPLICATION PROTOCOLS
TCP Connection Management Modeling. TCP Sliding Window. TCP congestion control. Introduction to application layer paradigms. Client Server model. Introduction and overview of HTTP protocol. Overview of FTP protocol. Operation of Electronic Mail. Introduction to peer-to-peer communication models. Introduction and overview of TELNET. Importance of Security in computer Networks.

TEXTBOOKS

1. Computer Networks - Andrew S Tanenbaum, 4th Edition, Pearson Education.
2. Data Communications and Networking - Behrouz A. Forouzan, Fifth Edition TMH, 2013.

REFERENCES

1. Computer Networking: A Top-Down Approach Featuring the Internet, James F. Kurose, K. W. Ross, 3rd Edition, Pearson Education.
2. Understanding communications and Networks, 3rd Edition, W. A. Shay, Cengage Learning.

SEMESTER-V

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 303 L	Computer Networks Lab	C	0	0	2	1

LIST OF PRACTICAL EXPERIMENTS

1.	Explain about wire shark and display how to send packets or packets from one layer to another.
2.	Write a Java program to implement Error Detection Technique using CRC Algorithm.
3.	Write a Java program to implement Error Correction Technique using Hamming code.
4.	Write a Java program to implement TCP Client Server programming.
5.	Write a Java program to implement UDP Client Server Programming.
6.	Write a Java program to implement 1-bit Stop and Wait Protocol at data link layer.
7.	Write a Java program to implement N-bit Sliding Window Protocol at data link layer.
8.	Write a Java program to implement Dijkstra Shortest path routing protocol.
9.	Write a Java program to implement Distance Vector Routing.
10.	Write a Java program to implement echo command in client server socket programming.
11.	Write a Java program to implement Trace-route command.
12.	Write a Java program to implement Ping command.
13.	Write a Java program to display the class of IP address, network mask and generate the subnet IP address based on the subnet bits entered from the keyboard.
14.	Write a Java program to implement sliding window protocol at the transport layer.
15.	Write a Java program to transfer file using TCP?

SEMESTER-V

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 306	Compiler Design	C	3	0	0	3

UNIT I: INTRODUCTION TO COMPILERS

Translators-Compilation and Interpretation-Language processors -The Phases of Compiler-Errors Encountered in Different Phases-The Grouping of Phases-Compiler Construction Tools - Programming Language basics.

UNIT II: LEXICAL ANALYSIS

Need and Role of Lexical Analyzer-Lexical Errors-Expressing Tokens by Regular Expressions-Converting Regular Expression to DFA- Minimization of DFA-Language for Specifying Lexical Analyzers-LEX-Design of Lexical Analyzer for a sample Language.

UNIT III: SYNTAX ANALYSIS

Need and Role of the Parser-Context Free Grammars -Top-Down Parsing -General Strategies-Recursive Descent Parser Predictive Parser-LL (1) Parser-Shift Reduce Parser-LR (0) Item-Construction of SLR Parsing Table -Introduction to LALR Parser - Error Handling and Recovery in Syntax Analyzer-YACC-Design of a syntax Analyzer for a Sample Language.

UNIT IV: SYNTAX DIRECTED TRANSLATION & RUN TIME ENVIRONMENT

Syntax directed Definitions-Construction of Syntax Tree-Bottom-up Evaluation of S-Attribute Definitions- Design of predictive translator - Type Systems-Specification of a simple type of checker-Equivalence of Type Expressions-Type Conversions. Intermediate code generation: Quadruples, Triples, Indirect triples, 3-address code RUN-TIME ENVIRONMENT: Source Language Issues-Storage Organization-Storage Allocation- Parameter Passing-Symbol Tables-Dynamic Storage Allocation-Storage Allocation in FORTAN.

UNIT V: CODE OPTIMIZATION AND CODE GENERATION

Principal Sources of Optimization-DAG- Optimization of Basic Blocks-Global Data Flow Analysis-Efficient Data Flow Algorithms-Issues in Design of a Code Generator - A Simple Code Generator Algorithm.

TEXTBOOKS

1. Compilers – Principles, Techniques and Tools, Alfred V Aho, Monica S. Lam, Ravi Sethi and Jeffrey D Ullman, 2nd Edition, Pearson Education, 2007.

REFERENCES

1. Vassiliadis, Vassilis, et al. "D2. 3: Advanced compiler implementation." Center for Research and Technology Hellas, Tech. Rep 2016.
2. Cooper, Keith, and Linda Torczon. Engineering a compiler. Elsevier, 2011.
3. Charles N. Fischer, Richard. J. LeBlanc, “Crafting a Compiler with C”, Pearson Education, 2008.

WEB RESOURCES

1. https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-004-computation-structures-spring-2017/c11/
2. https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-035-computer-language-engineering-spring-2010/
3. https://web.stanford.edu/class/archive/cs/cs143/cs143.1128/

SEMESTER-V

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 306 L	Compiler Design Lab	C	0	0	2	1

LIST OF PRACTICAL EXPERIMENTS

Week 1: Language recognizer

1. Write a program in C that recognizes the following languages.
    a. Set of all strings over binary alphabet containing even number of 0's and even number of 1's.
    b. **Lab Assignment:** Set of all strings ending with two symbols of same type.

Week 2: Implementation of Lexical analyzer using C

2. Implement lexical analyzer using C for recognizing the following tokens:
    - A minimum of 10 keywords of your choice
    - Identifiers with the regular expression : letter(letter | digit)*
    - Integers with the regular expression: digit+
    - Relational operators: <, >, <=, >=, ==, !=
    - Storing identifiers in symbol table.
    - Using files for input and output.

Week 3: Introduction to LEX tool

3. Implement the following programs using Lex tool
    a. Identification of Vowels and Consonants
    b. count number of vowels and consonants
    c. Count the number of Lines in given input
    d. Recognize strings ending with 00
    e. Recognize a string with three consecutive 0's

Week 4: Implementation of lexical analyzer using LEX

4. Implement lexical analyzer using LEX for recognizing the following tokens:
    - A minimum of 10 keywords of your choice
    - Identifiers with the regular expression : letter(letter | digit)*
    - Integers with the regular expression: digit+
    - Relational operators: <, >, <=, >=, ==, !=
    - Ignores everything between multi line comments (/* …. */)
    - Storing identifiers in symbol table
    - Using files for input and output.

Week 5: Lexical Analyzer

5. **Lab Assignment:**

    Consider the following mini Language, a simple procedural high-level language, only operating on integer data, with a syntax looking vaguely like a simple C crossed with Pascal. The syntax of the language is defined by the following BNF grammar:

    <program> ::= <block>

<block> ::= { <variabledefinition> <slist> } | { <slist> }
<variabledefinition> ::= int<vardeflist>;
<vardeflist> ::= <vardec> | <vardec>, <vardeflist>
<vardec> ::= <identifier> | <identifier> [<constant>]
<slist> ::= <statement> | <statement>; <slist>
<statement> ::= <assignment> | <ifstatement> | <whilestatement> | <block> | <printstatement> | <empty>
<assignment> ::= <identifier> = <expression> | <identifier> [<expression>] = <expression>
<ifstatement> ::= <bexpression> then <slist> else <slist> endif | if <bexpression> then <slist> endif
<whilestatement> ::= while <bexpression> do <slist> enddo
<printstatement> ::= print (<expression>)
<expression> ::= <expression> <additionop> <term> | <term> | addingop> <term>
<bexpression> ::= <expression> <relop> <expression>
<relop> ::= < | <= | == | >= | > | !=
<addingop> ::= + | -
<term> ::= <term><mulitop> <factor> | <factor>
<multop> ::= * | /
<factor> ::= <constant> | <identifier> | <identifier> [<expression>] | (<expression>)
<constant> ::= <digit> | <digit> <constant>
<identifier> ::= <identifier> <letterordigit> | <letter>
<letterordigit> ::= <letter> | <digit>
<letter> ::= a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z
<digit> ::= 0|1|2|3|4|5|6|7|8|9
<empty> has the obvious meaning
Comments (zero or more characters enclosed between the standard C / Java style comment brackets /*...*/) can be inserted. The language has rudimentary support for 1-dimensional arrays. The declaration int a[3] declares an array of three elements, referenced as a[0], a[1] and a[2]. Note also that you should worry about the scoping of names.

A simple program written in this language is:

```
{ int a[3], t1, t2;
  t1 = 2; a[0] = 1; a[1] = 2; a[t1] = 3;
  t2 = -(a[2] + t1 * 6)/ a[2] -t1);
  if t2 > 5 then
    print(t2);
  else  {
    int t3;
    t3 = 99;
    t2 = -25;
    print(-t1 + t2 * t3); /* this is a comment on 2 lines */
  } endif
}
```

Design a Lexical analyser for the above language. The lexical analyser should ignore redundant spaces, tabs, and newlines. It should also ignore comments. Although the syntax specification states that identifiers can be arbitrarily long, you may restrict the length to some reasonable value.

Week 6: Recursive Descent Parser

6. Implement Recursive Descent Parser for the Expression Grammar given below.

E TE’

E’ → +TE’ | ϵ
T → FT’
T’→ *FT’ | ϵ
F → (E) | i

7. **Lab Assignment:** Construct Recursive Descent Parser for the grammar
   G = ({S, L}, {(,), a, ,}, {S → (L) | a ; L→ L, S | S}, S) and verify the acceptability of the following strings:
   i. (a,(a,a))
   ii. (a,((a,a),(a,a)))

   You can manually eliminate Left Recursion if any in the grammar.

Week 7: Predictive parser

8. Write a C program for the computation of FIRST and FOLLOW for a given CFG

Week 8: Predictive Parser

9. Implement non-recursive Predictive Parser for the grammar
   S -> aBa
   B -> bB | ε

	A	b	$
S	S→aBa		
B	B→ε	B→bB	

10. **Lab Assignment:** Implement Predictive Parser using C for the Expression Grammar

E → TE’
E’→ +TE’ | ε
T → FT’
T’→ *FT’ | ε
F → (E) | d

Week 9: Shift Reduce Parser

11. Implementation of Shift Reduce parser using C for the following grammar and illustrate the parser’s actions for a valid and an invalid string.

E→E+E
E→E*E
E→(E)
E→d

12. **Lab Assignment:** Implementation of Shift Reduce parser using C for the following grammar and illustrate the parser’s actions for a valid and an invalid string.
   S –> 0S0 | 1S1 | 2

Week 10: LALR Parser

13. Implement LALR parser using LEX and YACC for the following Grammar:

E E+T |T
E’ T*F | F
F (E) | d

14. **Lab Assignment:** Implement LALR parser using LEX and YACC for the following Grammar by specifying proper precedence for operators:

    E E+E | E-E | E*E | E/E | -E | (E) | digit

Week 11:Intermediate code generation

15. Generate quadruples for given arithmetic expression using LEX and YACC.

Week 12: Intermediate code generation

16. Generate 3-address code for if statement using LEX and YACC.
17. **Lab Assignment:** Generate 3-address code for while statement using LEX and YACC.

Week 13: Code optimization

18. Implement constant propagation and folding using C for a given set of intermediate instructions.
19. **Lab Assignment:** Write a program to eliminate dead code

Week 14: Code optimization

20. Write a program to eliminate common sub expressions
21. **Lab Assignment:** Write a program to perform loop unrolling

Week 15: Code Generation

22. Generate machine code from the abstract syntax tree generated by the parser. The following instruction set may be considered as target code. The following is a simple register-based machine, supporting a total of 17 instructions. It has three distinct internal storage areas. The first is the set of 8 registers, used by the individual instructions as detailed below, the second is an area used for the storage of variables and the third is an area used for the storage of program. The instructions can be preceded by a label. This consists of an integer in the range 1 to 9999 and the label is followed by a colon to separate it from the rest of the instruction. The numerical label can be used as the argument to a jump instruction, as detailed below. In the description of the individual instructions below, instruction argument types are specified as follows:

    R specifies a register in the form R0, R1, R2, R3, R4, R5, R6 or R7 (or r0, r1, etc.).
    L Specifies a numerical label (in the range 1 to 9999).
    V Specifies a “variable location” (a variable number, or a variable location pointed to by a register -see below).
    A Specifies a constant value, a variable location, a register, or a variable location pointed to by a register (an indirect address). Constant values are specified as an integer value, optionally preceded by a minus sign, preceded by a #symbol. An indirect address is specified by an @followed by a register. So, for example, an A-type argument could have the form 4 (variable number 4), #4 (the constant value 4), r4 (register 4) or @r4 (the contents of register 4 identifies the variable location to be accessed).

    The instruction set is defined as follows:

    LOAD A,R
        loads the integer value specified by A into register R.
    STORE R,V

stores the value in register R to variable V.

OUT R
: outputs the value in register R.

NEG R
: negates the value in register R.

ADD A,R
: adds the value specified by A to register R, leaving the result in register R.

SUB A,R
: subtracts the value specified by A from register R, leaving the result in register R.

MUL A,R
: multiplies the value specified by Aby register R, leaving the result in register R.

DIV A,R
: divides register R by the value specified by A, leaving the result in register R.

JMP L
: causes an unconditional jump to the instruction with the label L.

JEQ R,L
: jumps to the instruction with the label L if the value in register R is zero.

JNE R,L
: jumps to the instruction with the label L if the value in register R is not zero.

JGE R,L
: jumps to the instruction with the label L if the value in register R is greater than or equal to zero.

JGT R,L
: jumps to the instruction with the label L if the value in register R is greater than zero.

JLE R,L
: jumps to the instruction with the label L if the value in register R is less than or equal to zero.

JLT R,L
: jumps to the instruction with the label L if the value in register R is less than zero.

NOP
: is an instruction with no effect. It can be tagged by a label.

STOP
: stops execution of the machine. All programs should terminate by executing a STOP instruction

SEMESTER-V

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 304	Database Management System	C	3	0	0	3

UNIT I: IINTRODUCTION TO DBMS AND RELATIONAL MODEL

File Processing System, Advantages of DBMS over File Processing System, Database System Applications DBMS Architecture: The three-schema architecture, Data Independence: Logical and Physical Data Models: Hierarchical, network and relation models, Introduction to relational model, concepts of domain, attribute, tuple, relation, importance of null values. Database constraints (Domain, Key constraints, integrity constraints) and their importance.

UNIT II: QUERY PROCESSING

Relational Algebra, Relational Calculus, Introduction to SQL: Database Objects- DDL Schema definitions. DML- Insert, select, update, delete. Views, exercise on SQL queries. Transaction support in SQL: Aggregate Functions, Null Values, Views, Complex Integrity Constraints in SQL, Assertions, Triggers.

UNIT III: CONCEPTUAL MODEL AND DATABASE DESIGN

Entity Relationship model Entity types, Entity Sets, Attributes, and Keys Relationships, Relationship types and constraints, Weak Entity types. Enhanced ER (EER) Modeling: Super/Sub Classes Specialization and Generalization. Constraints and characteristics of Specialization and Generalization. Example EER Schema. Basics of Normalization, Normal Forms: First Normal Form (1NF), Second Normal Form (2NF), Third Normal Form (3NF), BCNF, 4NF.

UNITIV: TRANSACTION PROCESSING, CONCURRENCY CONTROL AND RECOVERY

Introduction of transaction processing, advantages, and disadvantages of transaction processing system, Serializability and Recoverability of transaction, Concurrency Control, Lock based Protocols, Timestamp Based Protocols – Validation based Protocols - Multiple Granularity Locking, Recovery techniques.

UNIT V: OVERVIEW OF STORAGE AND INDEXING

Data on External Storage, File Organization, and Indexing - Clustered Indexes, Primary and Secondary Indexes.

Indexed Sequential Access Methods (ISAM) B+ Trees: Tree Structure, Search, Insert, Delete. Hash Based Indexing: Static Hashing, Extendable hashing, Linear Hashing, Extendible vs. Linear Hashing.

TEXTBOOKS

1. Ramez Elmasri and Shamkant Navathe. 2010. Fundamentals of Database Systems (6th ed.). Addison-Wesley Publishing Company, , USA.

REFERENCES

1. R. Ramakrishnan, J. Gehrke, Database Management Systems, McGraw Hill, 2004.
2. A. Silberschatz, H. Korth, S. Sudarshan, Database system concepts, 5/e, McGraw Hill, 2008.
3. Database system Implementation: Hector Garcia-Molina Jeffrey D. Ullman Jennifer Widom, Prentice Hall, 2000.

4. C.J. Date. 2003. An Introduction to Database Systems (8 ed.). Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

SEMESTER-V

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 304 L	Database Management System Lab	C	0	0	2	1

LIST OF PRACTICAL EXPERIMENTS

Exercise-I

Create a data file to store records of the students (fields: rollno, name, branch,age). (ii) Sort the records of the file based on the rollno of the students. (iii) Perform external sorting procedure (based on the roll number) on two data files which store records of the students and store the result in to the third data file.

Exercise-II

Store student records (fields: rollno,name,branch,age) in a data file and perform linear search in the data file by reading *rollno* as input and then display the student details and display the time required to do this operation.

Exercise-III

Store student records (fields: rollno,name,branch,age) in a data file and build an index file by considering the *rollno* as the key.

i. Perform linear search in the index file by reading *rollno* as input and then display the student details by reading from the data file and display the time required to do this operation.

ii. Perform binary search in the index file (by sorting the index file based on the *rollno*) by reading *rollno* as input and then display the student details by reading from the data file and display the time required to do this operation.

Exercise-IV

Store student records (fields: rollno,name,branch,age) in a data file and build an index file by using binary search tree (*rollno* is used as the key).

i. Perform search in the index file by reading *rollno* as input and then display the student details by reading from the data file and display the time required to do this operation.

ii. Add and delete the student records from the data file and then perform corresponding modifications in the index file.

Exercise-V

Store student records (fields: rollno,name,branch,age) in a data file and build an index file by using hash table (*rollno* is used as the key here).

iii. Perform search in the index file by reading *rollno* as input and then display the student details by reading from the data file and display the time required to do this operation.

i. Add and delete the student records from the data file and then perform corresponding modifications in the index file.

Exercise-VI

Consider the following relations.

Suppliers (<u>sid:</u> integer, sname: string, address: string)

Parts (pid: integer, pname: string, color: string)
Catalog (sid: integer, pid: integer, cost: real)
The key fields are underlined, and the domain of each field is listed after the field name.
Therefore, sid is the key for Suppliers, pid is the key for Parts, and sid and pid together form the key for Catalog. The Catalog relation lists the prices charged for parts supplied by Suppliers.

Write SQL statements for the following.

a. Find the names of suppliers who supply some **red color** part.
b. Find the sids of suppliers who supply some **red color** part and having office located at 'Chennai'
c. Find the average cost of **red color** parts supplied by various suppliers.
d. Find the names of the supplier who is supplying most number of parts.
e. Find the sids of suppliers who supply every part.
f. Find the sids of suppliers who supply every red color part.
g. List the number of suppliers for each color of part.
h. Find the supplier who supplies the **red color** part at a cheaper rate.
i. For each color part, display the details of the suppliers who supply that part at a cheaper rate.
j. Display the names of the suppliers along with the number of parts supplied by them.
k. Find the details of the supplier who supplies the costliest part.
l. Display the names of the suppliers who are selling at least two parts.

Exercise-VII

A) Consider the COMPANY database schema shown in the figure.

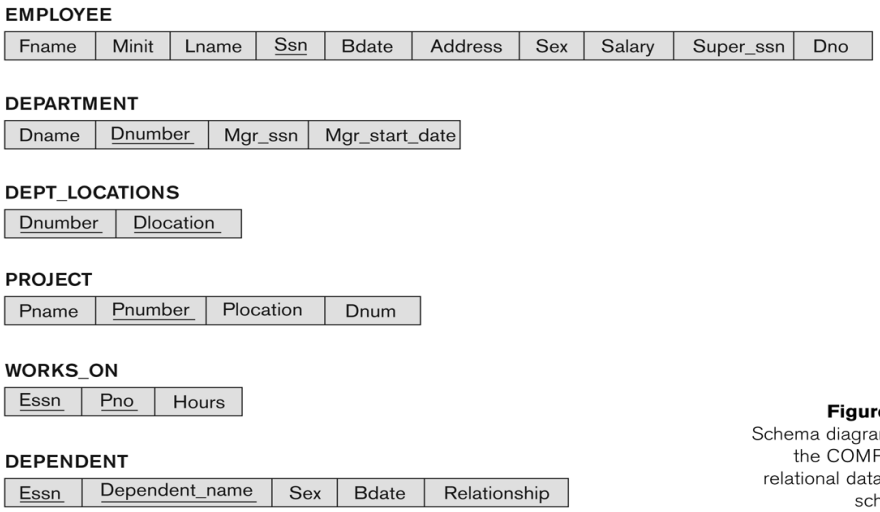

Figure 5.5
Schema diagram for the COMPANY relational database schema.

i. Create a view that has department name, manager name and manager salary for every department.
ii. Create a view that has project name, controlling depart name, number of employees, and total hours worked per week on the project for each project with more than one employee working on it.
iii. Create an updateable view for the relation DEPARTMENT

B) Create a materialized view for finding average salary of employees, average salary of managers, average salary for each department and department(s) which spend more money on salary for the employees.

C) Assume that Dno of EMPLOYEE relation has got NOT NULL constraint. Write a transaction which inserts tuples in to the relations EMPLOYEE and DEPARTMENT without affecting integrity constraints specified in the schema.

Exercise-VIII

A) Consider the following relations:
instructor(ID, name, dept_name, salary)
section(course_id, sec_ id, semester, year, building, room_ number, time_slot_id)
teaches(ID, course_id, sec_id, semester, year)
Write assertions for the following:

i. An instructor cannot teach in two different classrooms in a semester in the same slot
ii. An instructor cannot teach more than one course for the same semester

B) Consider the following relations.
product(maker, model, type)
pc(model, speed, ram, hd, price)
laptop(model, speed, ram, hd, screen , price)
printer(model, color, type, price)

Write triggers for the following:

(a) When updating the price of a PC, check that there is no lower priced PC with the same speed.
(b) When inserting a new printer, check that the model number exists in product.
(c) When making any modification to the Laptop relation, check that the average price of laptops for each manufacturer is at least Rs 1500.

C) Consider the following relations.
Emp (eno,ename,eage, salary,departno,supereno), dep(depno,depname,depage,eno), depart(departno,departname,location)

Write stored procedures

i. to find the average salary of employees who have got more than two dependents
ii. to find the names of employees (age is greater than 50) and their dependents (average age is less than 10).

Exercise-IX

Write java programs (using JDBC)

a. to create the following relations emp (eno,ename,eage, salary,departno,supereno), dep(depno,depname,depage,eno), depart(departno,departname,location) and insert at least 20 tuples for each relation.
b. (i) to find average age of employee's department wise (ii) to list department(s) (location wise) which pay less salary to the employees.

Exercise-X

Store student records (fields: rollno,name,branch,age) in a data file and build an index file by using B+ tree (rollno is used as the key here).

a. Perform search in the index file by reading roll no as input and then display the student details by reading from the data file and display the time required to do this operation.

b. Add and delete the student records from the data file and then perform corresponding modifications in the index file

SEMESTER-V

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 331	Industry Standard Coding Practice -3	ES	0	0	4	1

UNIT I

Problem solving with - Descriptive statistics, Mean/median/mode, Measures of dispersion/range variance, deviations, mean/median/mode problems, Random variables, Univariate & Bivariate random variables.

UNIT II

Problem solving with - Graphs, Handshaking Lemma, Simple Graphs, DFS/BFS, connected components, coloring, Introduction to DAGs, Spanning Trees, Articulation Points/ Connected points.

UNIT III

Problem solving with - Greedy Methods: Coin change, Fractional Knapsack, Activity Selections/ Job sequencing with Deadlines, Spanning Trees, Dynamic Programming: 0/1 Knapsack, Substructures, Longest common substring/subsequence, Longest Increasing sub sequence, Grid based Problems.

UNIT IV

Problem solving with - Divide & Conquer Strategies: Quick/Merge Sort, Min/Power functions, Backtracking, N Queens problem, Finding the path & Grid based problems, iterative/loop free approaches.

UNIT V

R Language Constructs, calculations, Operators, vectors, lists, Practice problems implementing R language, Matrices and data frame, Conditional statements and loops, Problem Solving on R language examples.

TEXTBOOKS

1. An Introduction to Statistical Learning: with Applications in R - Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani.
2. Introduction to Algorithms by Thomas H. Corman, The MIT Press, 3rd Edition.
3. Introduction to Algorithms: A Creative Approach by Udi Mander, Pearson.
4. R Cookbook - Paul Teetor, O'reilly.
5. Competitive Programming – 3e – Steven Halim, Felix Halim.

SEMESTER-V

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
ISES 301	Industry Specific Employability Skills-V	HS	1	1	0	0

UNIT I: QUANTS

Advanced Algebra, Advanced P & C and Probability, Advanced Time, Speed and Distance, Advanced Time and Work, Advanced Geometry and Mensuration.

UNIT II: COMMUNICATION SKILLS

Group discussion, Tell about yourself, Extempore, Mock interview, Video interview & Presentations

UNIT III: REASONING

Puzzle and Reasoning

TEXTBOOKS/REFERENCES

1. Mitchell S. Green – 2017, Know Thyself: The Value and Limits of Self-Knowledge.
2. Debbie Hindle, Marta Vaciago Smith - 2013 , Personality Development: A Psychoanalytic Perspective.
3. Lani Arredondo - 2000, Communicating Effectively.
4. Patsy McCarthy, Caroline Hatcher - 2002, Presentation Skills: The Essential Guide for Students.
5. Martha Davis, Elizabeth Robbins Eshelman, Matthew McKay - 2008, Time Management and Goal Setting: The Relaxation and Stress.
6. Arun Sharma – How to prepare for Quantitative Aptitude, Tata Mcgraw Hill.
7. RsAgarwal,A Modern Approach to Verbal and Non Verbal Reasoning,S.Chand Publications.
8. Verbal Ability and Reading comprehension-Sharma and Upadhyay.
9. Charles Harrington Elstor, Verbal Advantage: Ten Easy Steps to a Powerful Vocabulary, Large Print, September 2000.
10. GRE Word List 3861 – GRE Words for High Verbal Score, 2016 Edition.
11. The Official Guide to the GRE-General Revised Test, 2nd Edition, Mc Graw Hill Publication
12. English grammer and composition – S.C. Gupta.
13. R.S. Agarwal – Reasoning.
14. Reasoning for competitive exams – Agarwal.

SEMESTER V

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 340	UROP	PR	0	0	6	3

Department of Computer Science and Engineering, UROP Guidelines

1. Tentative date of commencement of Research Project is along with 6th semester each year.
2. The duration of the Project is 12 weeks or end by the 6th semester.
3. Maximum of 5 students form a team
4. Each faculty co-ordinates maximum 5teams
5. The title of the research work, scope, methodology and expected outcomes need to be approved by the Faculty mentor/guide.
6. Grading has to be completed by the concerned faculty by the end of 6th semester.
7. Number pf credits for CSE340 is 3.

General Guidelines for UROP project report and Research work.

These guidelines explain briefly the mechanics of writing a research paper in Computer Science and Engineering. These guidelines are generic and can be customized to fit most of the research works

The writing can start with the abstract, which can be approximately one page 10–20 sentences. The abstract will be refined and updated as a continuous process. The abstract can concisely (1) identify the research topic, (2) identify the benefits and advantages that result (3) and if there is novelty, describe the novelty of the presented work.

Section 1: Introduction (Motivation)

Although the title of the starting section is "Introduction" it should really be Motivation. In one or two paragraphs, the topic has to be introduced. This is followed with useful of the work, including possible applications of the work. Possible points to mention include:

1. Does the research work describe the state-of-the-art in that research domain?
2. What is the relevance of this work in filling any research gap?
3. Who will potentially benefit from the work?
4. Does the presented work provide a new technique of some sort?
5. Does this research work provide any new insight in some way?
6. Is it a review work which gives an insight to the current research in a particular domain?

Words like, contribute, benefit, advantageous, and possibly novel are used in this list. The presented work often builds on a previous system or algorithm. If so, your work may inherit benefits from the previous work. Those inherited advantages may also be listed.The introduction section then concludes with how the rest of the research paper is organized.

Section 2: Related Works: Presents review of the previous work on this topic.

The related work section demonstrates to the reader that you have done your homework (research), reviewed the previous literature, and now are ready to present your contribution based what has been previously published. The review is confined to relevant and recent research works in the domain of the proposed research. One of the difficult aspects of the related work section is choosing the proper scope. There is some subjectivity in choosing which books or papers to refer to and also importantly, which previous literature not to refer to. This is something an advisor is able to help with.

Section 3: Presents the proposed work/experimental/simulation specifications.

Section 4: Presents any algorithms or procedures used.

Next section: Can represent an evaluation of the resultsand the

Last section: May present conclusions and future work.

Citations

Any figure, image, or equation that is taken from another source must be cited. Content and terminology from other sources must also be cited. For more information about citations and their use, see:

http://www.plagiarism.org/. Click on the "How to cite sources" link.

References should be accurate and complete, i.e., with page numbers etc. A paper without complete and correct references can leave a bad impression on the reader and detract from a paper's credibility.

Mark Distribution: (As per the Original Plan. May be reviewed)

1. Internal evaluation by Guide: 50 marks
2. External evaluation by a Committee: 50 marks
   (Project Report, Demonstration and Presentation)

SEMESTER-VI

SEMESTER-VI

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 305	Software Engineering	C	3	0	2	4

UNIT I: SOFTWARE PROCESS AND AGILE DEVELOPMENT

Introduction to Software Engineering, Software Process, Perspective and Specialized Process Models –Introduction to Agility-Agile Process-Extreme programming-XP Process.

UNIT II: REQUIREMENTS ANALYSIS AND SPECIFICATION

Software Requirements: Functional and Non-Functional, User requirements, System requirements, Software Requirements Document – Requirement Engineering Process: Feasibility Studies, Requirement's elicitation and analysis, requirements validation, requirements management-Classical analysis: Structured system Analysis, Petri Nets-Data Dictionary.

UNIT III: SOFTWARE DESIGN

Design process – Design Concepts-Design Model– Design Heuristic – Architectural Design - Architectural styles, Architectural Design, Architectural Mapping using Data Flow- User Interface Design: Interface analysis, Interface Design –Component level Design: Designing Class based components, traditional Components.

UNIT IV: TESTING AND MAINTENANCE

Software testing fundamentals-Internal and external views of Testing-white box testing - basis path testing-control structure testing-black box testing- Regression Testing – Unit Testing – Integration Testing – Validation Testing – System Testing And Debugging –Software Implementation Techniques: Coding practices-Refactoring-Maintenance and Reengineering-BPR model-Reengineering process model-Reverse and Forward Engineering.

UNIT V: PROJECT MANAGEMENT

Software Project Management: Estimation – LOC, FP Based Estimation, Make/Buy Decision COCOMO I & II Model – Project Scheduling – Scheduling, Earned Value Analysis Planning – Project Plan, Planning Process, RFP Risk Management – Identification, Projection - Risk Management-Risk Identification-RMMM Plan-CASE TOOLS.

TEXTBOOKS

1. Roger S. Pressman, Software Engineering – A Practitioner‟s Approach, Ninth Edition, Mc Graw-Hill International Edition, 2020.
2. Ian Sommerville, Software Engineering, Tenth Edition, Pearson Education Asia, 2015.

REFERENCES

1. Rajib Mall, Fundamentals of Software Engineering, Fifth Edition, PHI Learning Private Limited, 2018.
2. Pankaj Jalote, Software Engineering, A Precise Approach, Wiley India, 2010.

3. Kelkar S.A., Software Engineering, Third Edition, Prentice Hall of India Pvt Ltd, 2013.
4. Stephen R. Schach, Object-oriented Software Engineering, Tata McGraw-Hill Publishing Company Limited,2008.

WEB RESOURCES

1. https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-355j-software-engineering-concepts-fall-2005/lecture-notes/
2. https://web.stanford.edu/class/archive/cs/cs295/cs295.1086/

SEMESTER-VI

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE305 L	Software Engineering Lab	C	0	0	2	1

LIST OF PRACTICAL EXPERIMENTS

Week 1: Software Requirement Specification

1. Develop requirements specification for a given problem

Week 2: Data Flow Diagram (DFD)

2. Develop DFD Model (Level 0, Level 1 DFD and data dictionary) of the sample problem

Week 3: DFD and Structured chart

3. To perform the function-oriented diagram: DFD and Structured chart

Week 4: Use case Diagram

4. To perform the user's view analysis: Use case diagram

Week 5: Class Diagram

5. To draw the structural view diagram: Class diagram

Week 6: Object Diagram

6. To draw the structural view diagram: Class diagram, object diagram

Week 7: Package Diagram

7. To draw the structural view diagram: Package Diagram

Week 8: Sequence Diagram

8. To draw the structural view diagram: Sequence Diagram

Week 9: Interaction Overview Diagram

9. To draw the structural view diagram: Interaction Overview Diagram

Week 10: State-chart Diagram

10. To draw the behavioral view diagram: State-chart diagram

Week 11: Activity diagram

11. To draw the behavioral view diagram: Activity diagram

Week 12: Component diagram

12. To draw the implementation view diagram: Component diagram

Week 13: Deployment diagram

13. To draw the environmental view diagram: Deployment diagram

Week 14: Unit Testing

14. To perform various testing using the testing tool -unit testing

Week 15: Integration Testing

15. To perform various testing using the testing tool -integration testing

TEXTBOOKS

1. Roger S. Pressman, Software Engineering – A Practitioner‟s Approach, Ninth Edition, Mc Graw-Hill International Edition, 2020.
2. Ian Sommerville, Software Engineering, Tenth Edition, Pearson Education Asia, 2015.
3. Rajib Mall, Fundamentals of Software Engineering, Fifth Edition, PHI Learning Private Limited, 2018.
4. Pankaj Jalote, Software Engineering, A Precise Approach, Wiley India, 2010.
5. Kelkar S.A., Software Engineering, Third Edition, Prentice Hall of India Pvt Ltd, 2013.
6. Stephen R. Schach, Object-oriented Software Engineering, Tata McGraw-Hill Publishing Company Limited,2008
7. https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-355j-software-engineering-concepts-fall-2005/lecture-notes/
8. https://web.stanford.edu/class/archive/cs/cs295/cs295.1086/
9. Grady Booch, James Rumbaugh, Ivar Jacobson, Unified Modeling Language User Guide, The, 2nd Edition, 2016.
10. Dr.K.V.N.S. Prasad, "Software Testing Tools", 1st Edition, Dream tech, 2011.

SEMESTER-VI

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
ISES 302	Industry Specific Employability Skills-VI	HS	1	1	0	0

UNIT I: QUANTS
Advanced LR & DJ

UNIT II: COMMUNICATION SKILLS
Group discussion, Tell about yourself, Extempore, Mock interview, Video interview & Presentations

UNIT III: REASONING
Puzzle and Reasoning

TEXTBOOKS/REFERENCES

1. Mitchell S. Green – 2017, Know Thyself: The Value and Limits of Self-Knowledge.
2. Debbie Hindle, Marta Vaciago Smith - 2013 , Personality Development: A Psychoanalytic Perspective.
3. Lani Arredondo - 2000, Communicating Effectively.
4. Patsy McCarthy, Caroline Hatcher - 2002, Presentation Skills: The Essential Guide for Students.
5. Martha Davis, Elizabeth Robbins Eshelman, Matthew McKay - 2008, Time Management and Goal Setting: The Relaxation and Stress.
6. Arun Sharma – How to prepare for Quantitative Aptitude, Tata Mcgraw Hill.
7. RsAgarwal,A Modern Approach to Verbal and Non Verbal Reasoning,S.Chand Publications.
8. Verbal Ability and Reading comprehension-Sharma and Upadhyay.
9. Charles Harrington Elstor, Verbal Advantage: Ten Easy Steps to a Powerful Vocabulary, Large Print, September 2000.
10. GRE Word List 3861 – GRE Words for High Verbal Score, 2016 Edition.
11. The Official Guide to the GRE-General Revised Test, 2nd Edition, Mc Graw Hill Publication
12. English grammer and composition – S.C. Gupta.
13. R.S. Agarwal – Reasoning.
14. Reasoning for competitive exams – Agarwal.

SEMESTER-VIII

SEMESTER-VIII

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 462	Capstone Project	PR	0	0	30	15

Capstone Project Guidelines

Introduction

These guidelines are conceived as a set of procedures stating broad expectations from both students and mentors of the Capstone project which is part of the B.Tech CSE curriculum. These guidelines are intended to make the project work evaluation process easier, formal and more authentic. The Capstone project has to be sufficiently complex and feasible so as to be considered for 12 Credits. The evaluation of the project is done by a review panel comprising department faculty members and the review process is continuous. In the first review by the constituted panel, the project may be accepted or rejected or major/minor changes can be suggested.

Project Selection

Capstone project may be an in-campus project or can be mapped with internship carried out in the industry or the research internship carried out in the other premier Universities in India/Abroad.

In campus project: The idea for student's Project may be a proposal from a faculty member or student's own, or perhaps a combination of the two. The project has to be sufficiently complex and feasible. Students are advised to choose a project that involves a combination of sound background research, a solid implementation, or piece of theoretical work, and a thorough evaluation of the Project's output. Interdisciplinary Project proposals and innovative Projects are encouraged and more appreciable.

Mapping with any Internship:

a. Any type of internships can be carried out by the students in the 7^{th} and 8^{th} semester after getting the due approval from the Project coordinator and the Head of the department.
b. The internship period has to be a minimum of 10 weeks of duration in each semester and the students could have carried out the practical work for at least 180 hrs during this period.
c. The internship has to involve some Software/Hardware design and implementation component and/or research component and the complexity of this work is expected to match the requirements of Capstone Project work.

Mentor allocation process: Students can form a batch of 4 (5 may be allowed in exceptional cases on the discretion of the project coordinators) and select their mentor provided the Faculty member accepts them and the faculty member has less than the specified number projects under his/her mentorship.

Project Equipment: In case of deserving projects for limited financing of equipment, the students can approach the concerned university authorities following due procedure.

Meetings with Your Supervisor:

Instructions to students: You must make sure that you arrange regular meetings with your Mentor. The meetings may be brief once your project is under way, but your Mentor needs to know that your work is progressing. You are also expected to be contactable throughout the project. You should inform the Mentor your contact details and keep these updated if these change.

Instructions to Mentors: Mentors are advised to maintain a project dairy depicting attendance of student and progress of project.

Legal and Ethical Considerations: If a student wants to do some project with some company where their relatives or friends work, the details need to be disclosed to their mentor. The mentor has to report the same to the project coordinators for permission. Again, if a student doing internship with a company, the data, procedures/algorithms and software developed may be classified and may not be allowed to submit in the report. The students need to consider that before requesting mapping.

Project Report format: Format of the report is similar to the format of standard Journal papers published. (Abstract-Literature Survey – Methodology – Algorithms – Simulation – Results - explanation of results - Future work etc)

Project milestones and Assessment

Starting date of the project to be taken as the commencement date of the semester as per academic calendar. The students are expected to plan from the beginning for at least one research publication in a reputed journal.

Stage 1: Title, Scope of the project and Literature survey to be submitted within 2 weeks from the commencement of the project. In the first review by the constituted panel, the project may be accepted or rejected or major/minor changes can be suggested.

Stage 2: Methodology, Requirement analysis and Deliverables to be submitted within 4 weeks from the commencement of the project.

Stage 3: Algorithms, project design and implementation plan have to be submitted within 6 weeks of the commencement of the semester. Internal review will be conducted by the Mentor and this review has a weightage of 50%.

Stage 4: Project implementation to be done and demonstrate that the project meets the requirements and expectations.

Stage 5: The results need to be analyzed and if any fine tuning required is to be done.

Final evaluation by expert committee at the end of the semester.
and this evaluation has a weightage of 50%.

SPECIALIZATION STREAMS

SPECIALIZATION STREAMS

Artificial Intelligence and Machine Learning Stream

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 413	Artificial Intelligence	SE	3	0	0	3

UNIT I: INTRODUCTION

What is Artificial Intelligence, Foundations and History of Artificial Intelligence, Applications of Artificial Intelligence, Intelligent Agents, Structure of Intelligent Agents.

UNIT II: SEARCH

Introduction to Search, Searching for solutions, Uniformed search strategies, Informed search strategies, Local search algorithms and optimistic problems, Adversarial Search, current-best-hypothesis search, least commitment search.

UNIT III: KNOWLEDGE REPRESENTATION AND REASONING

Inference, Propositional Logic, Predicate Logic (first order logic), Logical Reasoning, Forward &Backward Chaining, Resolution; AI languages and tools - Lisp, Prolog, CLIPS.

UNIT IV: PROBLEM SOLVING

Formulating problems, problem types, Solving Problems by Searching, heuristic search techniques, constraint satisfaction problems, stochastic search methods.

UNIT V: LEARNING

Overview of different forms of learning, decision trees, rule-based learning, neural networks, reinforcement learning.

Game playing: Perfect decision game, imperfect decision game, evaluation function, minimax, alpha-beta pruning.

TEXTBOOKS

1. Stuart Russell, Peter Norvig, “Artificial Intelligence – A Modern Approach”, Pearson Education, Third Edition, Pearson Education, 2008.

REFERENCES

1. Elaine Rich and Kevin Knight, “Artificial Intelligence”, McGraw-Hill, 3rd edition, 2017.
2. E Charniak and D McDermott, “Introduction to Artificial Intelligence”, Pearson.

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 413 L	Artificial Intelligence Lab	SE	0	0	2	1

LIST OF PRACTICAL EXPERIMENTS

1. Week 1: Artificial Intelligence Problem identification and PEAS description.
2. Week 2: Introduction to AI programming Language PROLOG.
3. Week 3: Study of facts, objects, predicates and variables in PROLOG.
4. Week 4: Study of arithmetic operators, simple input/output and compound goals in PROLOG.
5. Week 5: Study of string operations in PROLOG. Implement string operations like substring,
6. string position, palindrome etc.
7. Week 6: Write a prolog program to implement all set operations (Union, intersection,
8. complement etc.
9. Week 7: Write a program for Usage of rules in Prolog.
   Create a family tree program to include following rules 1. M is the mother of P *if* she is a parent of P and is female 2. F is the father of P *if* he is a parent of P and is male 3. X is a sibling of Y *if* they both have the same parent. 4. Then add rules for grandparents, uncle-aunt, sister and brother. Based on the facts, define goals to answer questions related to
10. Week 8: Write programs for studying Usage of arithmetic operators in Prolog.
    Accept name of the student, roll no, his/her subject name, maximum marks and obtained marks in the subject. (Take marks of at least 6 subjects). Compute the percentage of a student. Display his result with other information.
    Accept department, designation, name, age, basic salary, house rent allowance (HRA) of an employee. Compute dearness allowance (DA) which is 15% of basic salary. Determine the gross salary (basic salary+HRA+DA) of the employee. Display all information of the employee (Generate Payslip).
11. Week 9: Implement a program for recursion and list in PROLOG.
12. Week 10: WAP for studying usage of compound object and list in Prolog.
    Write a program to maintain inventory items using a compound object:
    (i) Accept from user the details of at least 10 objects. (ii)Display from user the details of objects entered by user (2) Find and display odd and even numbers from a given input list.
13. Week 11: Write a prolog program to solve “Water Jug Problem”.
14. Week 12: Write a program to implement a monkey banana problem.
15. Week 13: Write a program to implement 8 Queens Problem.
16. Week 14: Write a program to solve traveling salesman problem.
17. Week 15: Write a program to solve water jug problem using LISP.

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 336	Machine Learning	SE	3	0	0	3

UNIT I

Introduction: Introduction to Machine Learning: Introduction. Different types of learning, Hypothesis space and inductive bias, Evaluation. Training and test sets, cross validation, Concept of over fitting, under fitting, Bias and Variance

Linear Regression: Introduction, Linear regression, Simple and Multiple Linear regression, Polynomial regression, evaluating regression fit.

UNIT II

Decision tree learning: Introduction, Decision tree representation, appropriate problems for decision tree learning, the basic decision tree algorithm, hypothesis space search in decision tree learning, inductive bias in decision tree learning, issues in decision tree learning, over fitting in decision tree and methods to avoid over fitting.

Instance based Learning: K nearest neighbour, theCurse of Dimensionality, Feature Selection: univariate, multivariate feature selection approach, missing values ratio, high correlation filter, low variance filter, feature selection using decision tree, Feature reduction Techniques: Principal Component Analysis, Linear Discriminate Analysis

Recommender System: Content based system, Collaborative filtering based

UNIT III

Probability and Bayes Learning: Bayesian Learning, Naïve Bayes, Python exercise on Naïve Bayes, Logistic Regression

Support Vector Machine: Introduction, the Dual formulation, Maximum margin with noise, nonlinear SVM and Kernel function, solution to dual problem

UNIT IV

Artificial Neural Networks: Introduction, Biological motivation, ANN representation, appropriate problem for ANN learning, Perceptron, multilayer networks and the back propagation algorithm

UNIT V

Ensembles: Introduction, Bagging and boosting, Random forest, Discussion on some research papers.
Clustering: Introduction, K-mean clustering, agglomerative hierarchical clustering, Python exercise on k-mean clustering

TEXTBOOKS

1. Machine Learning. Tom Mitchell. First Edition, McGraw- Hill, 1997.
2. Alpaydin, Ethem. Introduction to machine learning. MIT press, 2020.

REFERENCES

1. Kevin P. Murphy, “Machine Learning: A Probabilistic Perspective”, MIT Press, 2012.
2. Christopher Bishop, “Pattern Recognition and Machine Learning” Springer, 2007.

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 336 L	Machine Learning Lab	SE	0	0	2	1

LIST OF PRACTICAL EXPERIMENTS

1. Basic exercises on Python Machine Learning Packages such as Numpy, Pandas and matplotlib
2. Python exercise on Feature engineering, data visualisation
3. Programs on Covariance, Correlation, Covariance Matrix and Correlation Matrix
4. Implement Linear Regression and calculate sum of residual error
5. Program to implement different distance functions
6. Program to implement decision tree learning
7. Program to implement K nearest neighbour classifier
8. Program to implement Principle Component Analysis
9. Program to implement perceptron for different learning task
10. Programs to implement ADALINE and MADALINE for given learning task
11. Program to implement classification task using Support Vector machine
12. Programs to implement different Clustering algorithms

REFERENCE BOOKS

1. Swamynathan, Manohar. Mastering machine learning with python in six steps: A practical implementation guide to predictive data analytics using python. Apress, 2019.
2. Raschka, Sebastian. Python machine learning. Packt publishing ltd, 2015.

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 314	Digital Image Processing	SE	3	0	0	3

UNIT I

Introduction: Digital Image fundamentals: Image sampling and quantization, relationship between pixels, Image acquisition and Pre-processing: Intensity transformations and spatial filtering, some basic intensity transformation functions, Histogram processing, spatial filters for smoothing and sharpening.

UNIT II

Filtering in the Frequency Domain: basic filtering in the frequency domain, image smoothing and sharpening Image Restoration: Image restoration/degradation model, noise models, restoration in the presence of noise only, estimating the degradation function.

UNIT III

Image segmentation: Fundamentals, point, line detection, basic edge detection techniques, Hough transform, Thresholding, basic global thresholding, optimal thresholding using Otsu's method, multi-spectral thresholding, Region based segmentation, region growing, region splitting and merging.

UNIT IV

Color Image Processing: color models, Color transformation Image Compression: Fundamentals, Some basic compression methods Morphological Image Processing: Erosion and Dilation, opening and closing, thinning, skeletonisation.

UNIT V

Image Representation: Shape features (Region-based representation and descriptors), area, Euler's number, eccentricity, elongatedness, rectangularity, direction, compactness, moments, covex hull, texture features, color features. Object and Pattern Recognition: Pattern and pattern classes, Matching, minimum distance or nearest neighbor classifier, matching by correlation, Optimum statistical classifier, Neural network classifier.

TEXTBOOKS

1. R.C. Gonzalez, R.E. Woods, Digital Image Processing, 3rd Edition, Pearson Education

REFERENCES

1. S. Sridhar, Digital Image Processing, Oxford University Press, 2011.
2. Milan Sonka, Vaclav Hlavac and Roger Boyele, Image processing, analysis, and machine vision. 3e, Cengage Learning, 2014.
3. Computer Vision A modern approach, David A. Forsyth and Jeam Ponce, Pearson Education.

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 314 L	Digital Image Processing Lab	SE	0	0	2	1

LIST OF PRACTICAL EXPERIMENTS

1. Perform the following operations using library functions
    a. Read, Display and write any color image in other formats.
    b. Find RED, GREEN and BLUE plane of the color image.
    c. Convert color image into gray scale image and binary image.
    d. Resize the image by one half and one quarter.
    e. Image rotates by 45, 90 and 180 degrees.
2. Create black and white images (A) of size 1024x1024. Which consists of alternative horizontal lines of black and white? Each line is of size 128.
   Create black and white images (B) of size 1024x1024. Which consists of alternative vertical lines of black and white? Each line is of size128.Perform the following operations on Image A and Image B.
    a. Image addition of A and B
    b. Subtraction of A and B
    c. Multiplying Images of A and B
    a. Create a grayscale image of size 256x1024. Intensity of image should vary sinusoidal.
    b. Create a white image of size 256x256, with black box of size 58x58 at centre.
3. Develop programs for following intensity transformation operation on a gray scale image. Collect any gray scale image from any source. Process that image using these operations.
    a. Image negative
    b. Log transformation and inverse log transform: s = c log (1+r), c is a const, r ≥ 0. s is pixel intensity of output image, r is the pixel intensity of input image. Study the effect of constant c on the quality of output image.
    c. Power law transformation: Study the effect of different values of Gamma used in this transformation.
    d. Contrast stretching
    e. Gray level slicing
4. Develop programs for following spatial filtering operations on a gray scale image.
    a. Averaging: Implement averaging filtering operations for different window sizes and study their effect on the quality of output image. Write your observations on output image quality.
    b. Weighted averaging: Implement weighted averaging filtering operations for different window sizes and study their effect on the quality of output image. Write your observations on output image quality.
    c. Median filtering: Implement weighted averaging filtering operations for different window sizes and study their effect on the quality of output image. Write your observations on output image quality.
    d. Max filtering
    e. Min filtering

5. Take a gray scale image and add salt and pepper noise. Write programs for following operations and observe their outputs
    a. Linear smoothing or Image averaging
    b. Weighted averaging
    c. Median filtering. Compare the output quality among Image averaging and median filtering.
    d. Max filtering
    e. Min filtering
6. Write programs to perform following sharpening operations on a gray scale image
    a. Laplacian filter
    b. Filtering using composite mask
    c. Unsharp masking
    d. High boost filtering
    e. Filtering using first order derivative operators such as sobel and prewitt mask.
7. Write a program to improve contrast of an image using histogram equalization. The prototype of the function is as below:

    histogram_equalisation(input_Image, no_of_bins);

    The function should return the enhanced image. Consider two low contrast input images. Study the nature of the output image quality in each case by varying the number of bins.
8. Take a low contrast gray scale image (A) and a high contrast gray scale image (B). Write a program to improve the contrast of A with the help of image B using histogram specification or matching. The prototype of the function is as below:

    Histogram_sp(input_Image, specified_Iage, no_of_bins);

    The function should return the enhanced image.
9. Develop programs to implement frequency domain smoothing filters (Ideal, Butterworth and Gaussian) and apply these filters on a gray scale image.
    a. Compare/comment on the output of Ideal, Butterworth and Gaussian Low pass Filters having the same radii (cutoff frequency) value.
    b. Consider a suitable gray scale image and demonstrate the ringing effect on the output of Ideal low pass frequency domain filter.
    c. Compare the output of Butterworth low pass filters (order n=2) for different cut-off frequencies (5, 15, 30, 90, 120).
    d. Compare the output of Gaussian low pass filters for different cut-off frequencies (5, 15, 30, 90, and 120).
10. Develop programs to implement frequency domain sharpening/High pass filters (Ideal, Butterworth and Gaussian) and apply these filters on a gray scale image.
    a. Compare/comment on the output of Ideal, Butterworth and Gaussian High pass Filters having the same radii (cutoff frequency) value.
    b. Consider a suitable gray scale image and demonstrate the ringing effect on the output of Ideal high pass frequency domain filter.
    c. Compare the output of Butterworth high pass filters (order n=2) for different cut-off frequencies (5, 15, 30, 90, 120).
    d. Compare the output of Gaussian high pass filters for different cut-off frequencies (5, 15, 30, 90, and 120).
11. Develop program to add different types of noise in a gray scale image and write functions to implement following filters for image restoration in presence of these noises.
    a. Remove Salt and Pepper Noise

b. Minimize Gaussian noise
c. Median filter and Weiner filter

12. Write and execute program for image morphological operations erosion and dilation.
13. Implement Morphological smoothing using opening and closing
14. Develop program to implement point and line detection masks. Detect points and lines using these masks for a given gray scale image.
15. Develop programs for edge detection using different edge detection mask.
16. Develop programs to achieve image segmentation using
17. Basic Global thresholding
18. Optimal global thresholding or Otsu's thresholding
19. Given a set of coordinates as boundary pixels in an image. Write a program to implement Hough Transform for joining the points using different lines.
20. Given a MXN image. Write a program to find the Co-occurrence matrix for a given angle and distance. Compute the Co-occurrence matrix features.
21. Given a MXN image. Write a program to find the Local Binary Pattern profile of the given image.

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 412	Principles of Soft Computing	SE	3	0	0	3

UNIT I: INTRODUCTION TO SOFT COMPUTING, ARTIFICIAL NEURAL NETWORK (ANN)

Fundamentals of ANN, Basic Models of an artificial Neuron, Neural Network Architecture, learning methods, Terminologies of ANN, Hebb network, Supervised Learning Networks: Perceptron, Adaline, Madaline, Multi-Layer Perceptron, Feed forward Back propagation Network: back propagation learning, Learning Effect of Tuning parameters of the Back propagation.

UNIT II: RBF NETWORK, ASSOCIATIVE MEMORY

Auto, hetero and linear associative memory, network, Adaptive Resonance Theory: ART1, ART2, Introduction to Computer vision, Introduction to Convolutional neural network, popular architectures: Alex Net, Google Net, VGG Net.

UNIT III: FUZZY LOGIC

Fuzzy set theory: crisp sets, fuzzy sets, crisp relations, fuzzy relations, Fuzzy Systems: Crisp logic predicate logic, fuzzy logic, fuzzy Rule based system, Defuzzification Methods, Fuzzy rule-based reasoning.

UNIT IV: GENETIC ALGORITHMS

Fundamentals of genetic algorithms: Encoding, Fitness functions, Reproduction. Genetic Modeling: Cross cover, Inversion and deletion, Mutation operator, Bit-wise operators, Bitwise operators used in GA. Convergence of Genetic algorithm. Applications, Real life Problems Particle Swarm Optimization and its variants.

UNIT V

Hybrid Soft Computing Techniques Hybrid system, neural Networks, fuzzy logic and Genetic algorithms hybrids Genetic Algorithm based Back propagation Networks: GA based weight determination applications: Fuzzy logic controlled genetic Algorithms soft computing tools, Applications.

TEXTBOOKS

1. Principles of Soft Computing- S.N. Sivanandan and S.N. Deepa, Wiley India, 2nd Edition 2018.

REFERENCES:

1. Neuro Fuzzy and Soft Computing, J. S. R. JANG, C.T. Sun, E. Mitzutani, PHI.
2. Neural Networks, Fuzzy Logic, and Genetic Algorithm (synthesis and Application) S. Rajasekaran, G.A. Vijayalakshmi Pai, PHI.

Course Code	Course name	Course Category	CREDITS			
			L	T	P	C
CSE 412 L	Principles of Soft Computing Lab	SE	0	0	2	1

LIST OF PRACTICAL EXPERIMENTS

1. Write a Python Program to implement a perceptron. The input is your semester marks.
2. Write a python program to extend the exercise given above to implement Feed Forward Network. The inbuilt function should not be used.
3. Write a python program to implement Hebb Network. The inbuilt function should not be used.
4. Write a python program to implement Multilayer Perceptron. The inbuilt function should not be used.
5. Write a python program to implement any ANN with back propagation learning Algorithm.
6. Write a Python Program to implement ART1 and ART 2.
7. Write a python program to implement CNN.
8. Write a python Programming to realize the working principles of popular architectures such as Alex Net, Google Net and VGG Net.
9. Write python Program to realize Fuzzy Sets arithmetic.
10. Write a python Program to realize fuzzy relations.
11. Write a python program to realize a fuzzy rule of any popular problem(s).
12. Write a python program to realize a defuzzification scheme for the above exercise.
13. Write a python Program to reason the fuzzy rules in exercises 12 and 13.
14. Write a python program to realize various steps of Genetic Algorithms.
15. Write a Python Program to realize GA based back propagation Networks.
16. Write a Python Program to realize Fuzzy Controlled Genetic Algorithms.

Cyber Security Stream

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSES 337	Cryptography	SE	3	0	0	3

UNIT I

History and overview of cryptography, Classical Encryption Techniques: Symmetric Cipher Model, Substitution Techniques, Transposition Techniques, Rotor Machines, And Steganography

UNIT II

Stream Ciphers and Block Ciphers, Attacks on block ciphers, Block Cipher Principles, The Data Encryption Standard (DES), Block Cipher Design Principles, Group, Rings, Field, Polynomial Arithmetic, The Euclidean Algorithm, Finite Fields of the Form GF(2^n)

UNIT III

Advanced Encryption Standard (AES), Stream Ciphers, RC4, The Chinese Remainder Theorem, Public Key Cryptography and RSA Algorithm, Diffie-Hellman Key Exchange, Elliptic Curve Cryptography.

UNIT IV

Cryptographic Hash Functions: Applications of Cryptographic Hash Functions, Two Simple Hash Functions, Requirements and Security, Secure Hash Algorithm (SHA), SHA-3.

UNIT V

Introduction to Block Chain, Bitcoin basics, Smart Contracts, Blockchain development platforms and APIs, Blockchain Ecosystems, Ethereum, Distributed Consensus, Blockchain Applications

TEXTBOOKS/REFERENCES

1. Stallings, William. Cryptography and network security, Principle and Practice. Pearson Education India, 2017.
2. R. Stinson Cryptography, Theory and Practice (Fourth Edition Edition)
3. Handbook of Applied Cryptography by A. Menezes, P. Van Oorschot, S. Vanstone.
4. Melanie Swan, Blockchain, Blueprint for a new Economy, OReilly

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 337 L	Cryptography Lab	SE	0	0	2	1

LIST OF PRACTICAL EXPERIMENTS

1. Write a program take text file as an input and print word, character count and ascii value of each characters as output. (Hint: Use open(), read() and split()).
2. Write a encryption program: Input: computerscienceengineeringsrmuniversity Output: gsqtyxivwgmirgiirkmriivmrkwvqyrmzivwmxc Hint: key =4 (play with ascii value).
3. Raju send an encrypted message (cipher text) "PHHW PH DIWHU WKH WRJD SDUWB" to Rani. Can you build decryption process and find out what is the message (plain text) send to Rani? Hint: try all keys.
4. Raju send encrypted message "ZICVTWQNGKZEIIGASXSTSLVVWLA" to Rani. Can you build decryption process and find out what is the message send to Rani. Hint: try all keys for each character.
5. Kohli have plain text "wewishtoreplaceplayer". Can you build encryption process and find out what is the cipher text he needs send to BCCI. Help him out by using monoalphabatic cipher. Hint: use any one-to-one mapping between alphabets.
   One to one mapping

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	N	D	R	E	W	I	C	K	S	O	H	T	B	F	G	J	L	M	P	Q	U	V	X	Y	Z

6. Kohli sent encrypted message (Cipher text) "SEEMSEAOMEDSAMHL" to Anushka. Can you build decryption process and find out what is the message (plain text) send to Anushka. Hint: use above one to one mapping between alphabets.
7. Raju want to build encrypted and decryption algorithms of Playfair Cipher. Help him to build a key matrix using the key "srmapuniversity"
8. By using key matrix Raju want to send message "we are discovered save yourself" to Rani. Can you build encryption process and find out what is the cipher text message send to Rani by using palyfaircipher.

s	r	m	a	p
u	n	i/j	v	e
t	y	b	c	d
f	g	h	k	l
o	q	w	x	z

9. By using key "CBDE" Raju would like send message (plain text)"HELLO WORLD" to Rani. Can you build encryption process and find out what is the encrypted message (cipher text) to Raju by using Hill Cipher.Also Can you build decryption process and find out what is the decrypted message (plain text) of cipher text "SLHZYATGZT" by using Hill Cipher.
10. Implementation of Encryption and Decryption of Vigenère Cipher
    keyword *deceptive*
    key: deceptivedeceptivedeceptive
    plaintext: wearediscoveredsaveyourself
    ciphertext: ZICVTWQNGRZGVTWAVZHCQYGLMGJ
11. Implement the Encryption and Decryption of Row Transposition.

Key: 4 3 1 2 5 6 7
Plaintext: a t t a c k p
o s t p o n e
d u n t i l t
w o a m x y z
Ciphertext: TTNAAPTMTSUOAODWCOIXKNLYPETZ

12. Implement the Euclidean Algorithm for integers and polynomials.
13. Implement AES Key Expansion.
14. Implementation of AES encryption and decryption
15. Implementation of Simplified DES Encryption and decryption
16. Implementation of RC4
17. Implementation of RSA algorithm
18. Implementation of Diffie-Helman key exchanges
19. Implementation of elliptic-curve cryptography
20. Implementation of Hash functions

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 315	Network Security	SE	3	0	0	3

UNIT I: NEED FOR SECURITY

Need for Security: Security Attack, Security Services, Information Security, Methods of Protection.

Network Concepts: Basic Concepts of Computer Networks

Threats in Networks: Threat Precursors, Threats in Transit, Protocol Flaws, Message Confidentiality Threats, Non-existent and Well-Known Authentication, Spoofing, DoS, DDoS

Network Security Controls: Segmentation, Redundancy, Single Points of Failure, Encryption, Link and End-to-End Encryption, Virtual Private Networks, VPN & Firewall, PKI and Certificates, SSL and SSH Encryption, Kerberos, Onion Routing

UNIT II: AUTHENTICATION

Message Authentication Codes (MAC): Message Authentication Requirements, Message Authentication Functions, Security of MACs, MACs Based on Hash Functions: HMAC.

Digital Signature: Digital Signatures, Elgamal Digital Signature Scheme, Schnorr Digital Signature Scheme, NIST Digital Signature Algorithm, Elliptic Curve Digital Signature Algorithm, RSA-PSS Digital Signature Algorithm.

Overview of Authentication Systems: Password-Based Authentication, Address-Based Authentication, Cryptographic Authentication Protocols, Trusted Intermediaries, KDCs, Certification Authorities (CAs), Session Key Establishment.

Security Handshake Pitfalls: Login, Mutual Authentication, Integrity/Encryption for Data, Two-Way Public Key Based Authentication, One-Way Public Key Based Authentication, Mediated Authentication (with KDC), Needham-Schroeder, Expanded Needham-Schroeder, Otway-Rees, Nonce Types.

Strong Password Protocols: Lamport's Hash, Strong Password Protocols, Strong Password Credentials Download Protocols.

UNIT III: IPSEC

IPSec: Overview of IP Security (IPSec), IP Security Architecture, Modes of Operation, Security Associations (SA), Authentication Header (AH), Encapsulating Security Payload (ESP), Comparison of Encodings.

Internet Key Exchange (IKE): Photuris, SKIP, History of IKE, IKE Phases, Phase 1 IKE - Aggressive Mode and Main Mode, Phase 2/Quick Mode, Traffic Selectors, The IKE Phase 1 Protocols, Phase-2 IKE: Setting up IPsec SAs, ISAKMP/IKE Encoding - Fixed Header, Payload Portion of ISAKMP Messages, SA Payload, SA Payload Fields.

UNIT IV: WEB SECURITY

Web Security Requirements: Web Security threats, Web traffic Security Approaches.

SSL/TLS: Secure Socket Layer (SSL), Transport Layer Security (TLS), TLS Architecture, TLS record protocol, change cipher spec protocol, Alert Protocol, Handshake Protocol, Https, SSH.

Secure Electronic Transaction (SET): SET functionalities, Dual Signature, Roles & Operations, Purchase Request Generation, Purchase Request Validation, Payment Authorization and Payment Capture.

SNMP: Basic concepts of SNMP, SNMP basic components and their functionalities, Basic commands of SNMP, SNMPv1 Community facility and SNMPv3. Intruders, Viruses and related threats.

UNIT V: FIREWALL & EMAIL SECURITY

Firewalls: Need for Firewalls, Firewall Characteristics, Types of Firewalls, Firewall Basing, Firewall Location and Configurations.

Electronic Mail Security: Pretty Good Privacy, S/MIME, DNSSEC, Domain Keys Identified Mail.

TEXTBOOKS

1. Perlman, Radia, Charlie Kaufman, and Mike Speciner. Network security: private communication in a public world. Pearson Education India, 2016.
2. Cryptography and Network Security – Principles and Practice: William Stallings, Pearson Education, 6th Edition.

REFERENCES

1. Network Security and Cryptography, Bernard Menezes, CENGAGE Learning.
2. Introduction to Network Security: Neal Krawetz, CENGAGE Learning.
3. Cryptography and Network Security: Atul Kahate, Mc Graw Hill, 3rd Edition.

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSES 315 L	Network Security Lab	SE	0	0	2	1

LIST OF PRACTICAL EXPERIMENTS

1. packet assembler/analyzer: Wireshark.
2. packet assembler/analyzer: hPing3.
3. Encrypted communication over socket using AES.
4. Message Authentication Code: MAC.
5. MAC Based on Hash Function: HMAC.
6. Session Key establishment using RSA.
7. Handcraft a TCP handshake.
8. Diffie-Hellman Algorithm.
9. DH Key exchange.
10. Network Mapper: Nmap Basics.
11. Penetration Testing: Metasploit Basics.
12. Key tool & OpenSSL.
13. One Way SSL to a Web App.
14. SNMP: net SNMP – MIB.
15. Firewall with UFW.

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 410	Mobile and Wireless Security	SE	3	0	0	3

UNIT I: INTRODUCTION TO MOBILE AND WIRELESS NETWORKS

IEEE wireless networks, WLAN: IEEE 802.11 (a:n), WPAN: IEEE 802.15 (Bluetooth & Zigbee), WMAN: IEEE 802.16 (WiMAX), WMAN mobile: IEEE 802.20 (MBWA), IEEE 802.21 framework (MIH), Cellular Networks, Cellular networks: VoIP, IMS, 4G Security

UNIT II: HOW EXISTING WIRELESS NETWORKS ARE SECURED

Attacks on wireless networks, WEP, WEP Shortcomings, IEEE 802.11i, Bluetooth, Authentication in wireless networks, GSM Authentication, UTMS Authentication, SS7 Protocol Stack

UNIT III: NEXT GENERATION WIRELESS NETWORKS

Mobility & Internet, Mobility with MIPv6, Mobility with Mobile IPv4, IP mobility with HIP and NetLMM, Ad Hoc Networks: Protocols, Security in Ad Hoc Networks, Key Management in Ad Hoc Networks, Wireless Sensor Network Security, Key Management in WSN

UNIT IV: PREVENTING MALICIOUS BEHAVIOR

Naming and Addressing, Establishing Security Association: Key Establishment in Sensor Network, Establishing Security Association: Utilizing Mobility, Wormhole Attack, Privacy in RFID System, Location Privacy in Vehicular Network, Privacy Preserving Routing in Ad-hoc Networks

UNIT V: MOBILE APPLICATION SECURITY

Brief Introduction to Android – I, Brief Introduction to Android – II, Android Security Model Permission, Package Management, User Management, Cryptographic Providers, Network Security and PKI, Credential Storage, Discovering Vulnerabilities using Static Analysis, Tools Fuzzing on Android.

TEXTBOOKS

1. Noureddine Boudriga, Security of Mobile Communications, 2010.
2. Levente Buttyán and Jean-Pierre Hubaux, Security and Cooperation in Wireless Networks, 2008. [Available Online]

REFERENCES

1. James Kempf, Wireless Internet Security: Architectures and Protocols, 2008.
2. Android Security Internals: An In-Depth Guide to Android's Security Architecture, Author: Nikolay Elenkov, No Starch Press, First Edition, Nov. 2014

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 410 L	Mobile and Wireless Security Lab	SE	0	0	2	1

LIST OF PRACTICAL EXPERIMENTS

1. Understanding IEEE 802.11with Wireshark.
2. Medium Access Control for Wirelessly Connected Stations.
3. Wireless Security – I (Wireless Security Basics).
4. Wireless Security – II (Wireless Threats).
5. Bluetooth Security.
6. Wireless Security Pen Testing (WEP, WPA/WPA2).
7. Mobility & Load and Congestion Window Size.
8. server mobility on the network performance: Load (bits/sec) , Congestion Window Size. (bytes) , and Traffic Received (bytes).
9. Queuing Disciplines and VoIP.
10. Network Security and Virtual Private Networks.
11. Network Application Performance Analysis.
12. Connection-Oriented, Cell-Switching Technology.
13. Developing Android App.
14. Reverse Engineering using Apktool and dex2jar.
15. Analyzing Vulnerabilities using Static Analyzer and Fuzzer.

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 414	Internet Protocols and Networking	SE	3	0	0	3

UNIT I

Network Models: Layered Tasks, The OSI Model, Layers in OSI Model, TCP/IP Protocol suite, Addressing. Connecting devices: Passive Hubs, Repeaters, Active Hubs, Bridges, Two Layer Switches, Routers, Three Layer Switches, Gateway, Backbone Networks.

UNIT II

Principles of Internetworking, Connectionless Interconnection, Application-Level Interconnection, Network Level Interconnection, Properties of the Internet, Internet Architecture, Interconnection through IP Routers TCP, UDP & IP: TCP Services, TCP Features, Segment, A TCP Connection, Flow Control, Error Control, Congestion Control, Process to Process Communication, User Datagram, Checksum, UDP Operation, IP Datagram, Fragmentation, Options, IP Addressing: Classful Addressing, IPV6.

UNIT III

Transport layer Protocols: Transport Layer Services, UDP and TCP protocols, Flow control and Error control in Transport layer, Flow control mechanisms in Transport layer.

UNIT IV

Data Traffic, Congestion, Congestion Control, Congestion Control in TCP, Congestion Control in Frame Relay, Source Based Congestion Avoidance, DEC Bit Scheme, Quality of Service, Techniques to Improve QOS: Scheduling, Traffic Shaping, Admission Control, Resource Reservation, Integrated Services and Differentiated Services.

UNIT V

Concepts of Buffer Management, Drop Tail, Drop Front, Random Drop, Passive Buffer Management Schemes, Drawbacks of PQM, Active Queue Management: Early Random Drop, RED Algorithm.

TEXTBOOKS/REFERENCES

1. Douglas. E. Comer, “Internetworking with TCP/IP “, Volume I PHI.
2. Behrouz A Forouzan, “TCP/IP Protocol Suite”, TMH, 3rd Edition.
3. B.A. Forouzan, “Data communication & Networking”, TMH, 4th Edition.

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 414 L	Internet Protocols and Networking Lab	SE	0	0	2	1

LIST OF PRACTICAL EXPERIMENTS

1. Install and Configure Wired and Wireless NIC and transfer files between systems in LAN and Wireless LAN.
2. Study basic network command and network configuration commands.
3. Configure Host IP, Subnet Mask and Default Gateway in a System in LAN (TCP/IP Configuration).
4. Establish Peer to Peer network connection using two systems using Switch and Router in a LAN.
5. Configure a Network topology using Packet Trace.
6. Configure Internet connection and use IPCONFIG, PING / Tracer and Net stat utilities to debug the network issues.
7. Transfer files between systems in LAN using FTP Configuration, install Print server in a LAN and share the printer in a network.
8. Set up a network that utilizes TCP as its end-to-end transmission protocol, and analyse the size of the congestion window with different mechanisms.
9. Implement flow control so that a fast sender will not overrun a slow receivers' buffer.
10. Implement RED algorithm DEC Bit scheme in TCP.
11. Implement the Drop Tail Buffer Management Policies.
12. Implement the Drop Front Buffer Management Policies.
13. Implement the Random Drop Buffer Management Policies.
14. Implement the Early Random Drop Buffer Management Policies.
15. Implement RED algorithm.

Big Data and Analytics Stream

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 310	Data Warehousing and Mining	SE	3	0	0	3

UNIT I

Data warehousing and Online Analytical Processing: Basic concepts of Data Warehouse – Data Warehouse Modelling – Data Warehouse Design and Usage – Data Warehouse Implementation – Data Generalization by Attribute-oriented Induction.

UNIT II

Data Mining: Knowledge Discovery from Data – Types of Data - Data Mining Functionalities – Data Preprocessing – Data Cleaning – Data Integration – Data Reduction – Data Transformation and Data Discretization. Association Rule Mining – Frequent Itemset Mining methods – Pattern Evaluation Methods.

UNIT III

Classification – Basic Concepts – Decision Tree Induction – Bayes Classification Methods – Rule based Classification – Model Evaluation and Selection – Techniques to improve Classification Accuracy

UNIT IV

Clustering – Cluster Analysis – Partitioning Methods – Hierarchical Methods – Density-Based Methods – Grid Based Methods – Evaluation of Clustering.

UNIT V

Data Mining Trends and Research Frontiers - Mining Complex Data types – Other Methodologies of Data Mining – Data Mining Applications – Data Mining and Society – Data Mining trends.

TEXTBOOKS

1. Jiawei Han, Micheline Kamber and Jian Pei “Data Mining Concepts and Techniques”, Third Edition, Elsevier, 2011.

REFERENCES

1. G. K. Gupta “Introduction to Data Mining with Case Studies”, Third Edition, Prentice Hall of India, 2014.
2. Pang-Ning Tan, Michael Steinbach and Vipin Kumar “Introduction to Data Mining”, Pearson Education, 2016.
3. K.P. Soman, Shyam Diwakar and V. Ajay “Insight into Data mining Theory and Practice”, Easter Economy Edition, Prentice Hall of India, 2006.
4. Alex Berson and Stephen J. Smith “Data Warehousing, Data Mining & OLAP”, Tata McGraw – Hill Edition, Thirteenth Reprint 2008.

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 310 L	Data Warehousing and Data Mining Lab	SE	0	0	2	1

LIST OF PRACTICAL EXPERIMENTS

1. **Week 1:** Implementation of OLAP operations.
2. **Week 2:** Data pre-processing techniques.
3. **Week 3:** Write a program in any programming language to generate at least 10,000 transactions in a text file with at least three items.
4. **Week 4 & 5:** Write a program to implement the APRIORI algorithm.
5. **Week 6 & 7:** Write a program for FP-Growth algorithm.
6. **Week 8 & 9:** Write a program to implement Decision tree-based classification.
7. **Week 10 & 11:** Write a program to implement Bayesian classification.
8. **Week 12:** Write a program to implement K-means clustering.
9. **Week 13:** Write a program to implement Divisive clustering.
10. **Week 14:** Write a program to implement Agglomerative clustering.
11. **Week 15:** Write a program to implement DBSCAN clustering.

TEXTBOOKS

1. Jiawei Han, Micheline Kamber and Jian Pei “Data Mining Concepts and Techniques”, Third Edition, Elsevier, 2011.

REFERENCES

1. G. K. Gupta “Introduction to Data Mining with Case Studies”, Third Edition, Prentice Hall of India, 2014.
2. Pang-Ning Tan, Michael Steinbach and Vipin Kumar “Introduction to Data Mining”, Pearson Education, 2016.
3. K.P. Soman, Shyam Diwakar and V. Ajay “Insight into Data mining Theory and Practice”, Easter Economy Edition, Prentice Hall of India, 2006.
4. Alex Berson and Stephen J. Smith “Data Warehousing, Data Mining & OLAP”, Tata McGraw – Hill Edition, Thirteenth Reprint 2008.

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 338	Applied Data Science	SE	3	0	0	3

UNIT I: INTRODUCTION

Introduction to Data Science, Data vs. Big Data, Statistical Inference - Populations and samples, Statistical modeling, probability distributions, fitting a model. Data Science Process, Exploratory Data Analysis, Basic tools - plots, graphs and summary statistics of EDA. Introduction to R Programming.

UNIT II

Basic Machine Learning Algorithms - Linear Regression - K-Nearest Neighbors (K-NN) - K-means, K-Medoids, Naive Bayes. Case Study: Real Direct (online real estate firm), Filtering Spam - Linear Regression and K-NN and Naive Bayes for Filtering Spam. Data Wrangling: APIs and other tools for scrapping the Web - Feature Generation and Feature Selection (Extracting Meaning from Data) - Motivating Application and Case Study: User (customer) retention - Feature Generation - Feature Selection algorithms – Filters; Wrappers; Decision Trees; Random Forests.

UNIT III

Recommendation Systems: Building a User-Facing Data Product - Algorithmic ingredients of a Recommendation Engine - Dimensionality Reduction - Singular Value Decomposition - Principal Component Analysis.

UNIT IV

Mining Social-Network Graphs - Social networks as graphs - Clustering of graphs - Direct discovery of communities in graphs - Partitioning of graphs - Neighborhood properties in graphs.

UNIT V

Data Visualization - Basic principles, ideas and tools for data visualization – Case Study 1 on industry projects – Case Study 2: Create Complex visualization dataset - Data Science and Ethical Issues - Discussions on privacy, security, ethics - Next-generation data scientists.

TEXTBOOKS

1. Sinan Ozdemir, Sunil Kakade. Principles of Data Science - Second Edition Released December 2018 Publisher(s): Packt Publishing ISBN: 9781789804546.
2. Cathy O'Neil and Rachel Schutt Doing Data Science, Straight Talk from The Frontline. O'Reilly. 2014.

REFERENCES

1. Jure Leskovek, Anand Rajaraman and Jeffrey Ullman Mining of Massive Datasets v2.1, Cambridge University Press 2014 (free online).
2. Kevin P. Murphy. Machine Learning: A Probabilistic Perspective. ISBN 0262018020. 2013.
3. Foster Provost and Tom Fawcett. Data Science for Business: What You Need to Know about Data Mining and Data-analytic Thinking. ISBN 1449361323. 2013.
4. Trevor Hastie, Robert Tibshirani and Jerome Friedman Elements of Statistical Learning, Second Edition ISBN 0387952845 2009 (free online).
5. Avrim Blum, John Hopcroft and Ravindran Kannan Foundations of Data Science (Note: this is a book currently being written by the three authors. The authors have made the first draft of their notes for the book available online. The material is intended for a modern theoretical course in computer science.)

6. Mohammed J. Zaki and Wagner Miera Jr. Data Mining and Analysis: Fundamental Concepts and Algorithms. Cambridge University Press. 2014.
7. Jiawei Han, Micheline Kamber and Jian Pei Data Mining: Concepts and Techniques, Third Edition. ISBN 0123814790 2011.

Course Code	Course Name	Course Category	Credits			
			L	T	P	C

CSE 338 L	Applied Data Science Lab	SE	0	0	2	1

LIST OF PRACTICAL EXPERIMENTS

1. Write a python program to apply datafication concepts of friendship network of your face book account.
2. Write python program to calculate the central tendency of any popular data set. The inbuilt functions in the python should not be used.
3. Write R – Programming to plot various charts and graphs. You have to consider minimum two popular data sets and draw all the statistical observations.
4. Write a python Program to apply EDA on any two popular data sets and provided your analysis and interpretations. Use matplotlib library of python along with other libraries for the analysis and interpretation.
5. Write Python program to implement Linear Regression using inbuilt python Library. Also, write your own program to implement Linear Regression without using the inbuilt function. Compare and contrast the results.
6. Write Python program to implement K-Nearest Neighbors using inbuilt python Library. Also, write your own program to implement K-Nearest Neighbors without using the inbuilt function. Compare and contrast the results.
7. Write Python program to implement K-Means using inbuilt python Library. Also, write your own program to implement K-Means without using the inbuilt function. Compare and contrast the results.
8. Write a python program to implement a Spam Filter using Linear Regression and K-NN. Use a popular dataset.
9. Write a Python Program to Scrapping the Web using suitable API. Create a usable dataset for classification and clustering purpose.
10. Write a python program to generate the features from the data set created by you for exercise 9.
11. Write a Python Program to implement Filter and Wrappers.
12. Write a Python Program to implement Decision Trees, Random Forests – The inbuilt functions should not be used for the implementation.
13. Write a python Program to implement Singular Value Decomposition and Principal Component Analysis. Use any popular data set.
14. Write a python Program to extract the friendship details of your face book account as Social network Graph and represent in various visual forms.
15. Write a python program to extend the above exercise to discover the communities in the graph, partition the graph and extracting the neighbor hood properties of the graphs.
16. Write Python Program using Bokeh 2.1.1 realize the all the basic principles of data visualization.
17. Consider any popular dataset and present complex visualization principle using Bokeh 2.1.1.

Course Code	**Course Name**		**Credits**

		Course Category	L	T	P	C
CSE 417	Principles of Big Data Management	SE	3	0	0	3

UNIT I

Understanding Big Data – Concepts and Terminology – Big Data Characteristics – Different types of Data – Big Data Storage concepts – Clusters – File systems and distributed file systems – NoSQL – Sharding – Replication – CAP theorem – BASE - Hadoop Distributed File System (HDFS) Architecture - HDFS commands for loading/getting data - Accessing HDFS through Java program.

UNIT II

Big Data Processing Concepts – Parallel Data Processing – Distributed Data Processing – Hadoop – Processing workloads – Batch processing with MapReduce – Map and Reduce Tasks – MapReduce Example

UNIT III

Hadoop ecosystem and its components– Flume - Sqoop - Pig - Spark - Hbase.

UNIT IV

Querying big data with Hive: Introduction to Hive QL - Hive QL: data definition- data manipulation – Hive QL Queries.

UNIT V

Data Analytics using R: Introduction to R, Creating a dataset, Getting started with graphs, Basic data management, Advanced data management.

TEXTBOOKS/REFERENCES

1. Big Data Fundamentals: concepts, Drivers and Techniques: Person Education, 2016
2. Hadoop The Definitive Guide, IV edition, O'Reilly publications
3. Hadoop in Action, Chuck lam, Manning publications
4. Programming, Hive, O'Reily publications,
5. Apache Hive Cookbook, PACKT publications
6. R in Action, Robert I. Kabacoff, Manning publications
7. Practical Data Science with R, Nina Zumel John Mount, Manning publications.

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 417 L	Principles of Big Data Management Lab	SE	0	0	2	1

LIST OF PRACTICAL EXPERIMENTS

1.a. Hadoop Installation
   b. Hadoop Shell Commands
2.a. Writing a file from local file system to Hadoop Distributed file system (HDFS)
   b. Reading a file from HDFS to the local file system.
3.a. Implementation of Word Count program using MapReduce without combiner logic.
   b. Implementation of Word Count program using MapReduce with combiner logic**.**
4.Weather data analysis for analyzing hot and cold days using MapReduce.
5. Implementation of MapReduce algorithm for Matrix Multiplication.
6. Implement a MapReduce program to identify "common friends" among all pairs of users.
7. Transfer data between Hadoop and relational database servers using Sqoop.
8. Read a text file from HDFS into RDD using Spark.
9. Use HiveQL to analyze the stock exchange dataset and calculate the covariance between the stocks for each month. This will help a stock-broker in recommending the stocks to his customers.
10. Implement JOINS using HIVE
    a. Inner Join
    b. Left outer join
    c. Right outer Join
    d. Full outer join

11. Write a R program to create a student record using the Vector concept.
12. Write a R program to create medical patients status using data frame
        i) Patient age ii) Gender iii) Symptoms iv) Patient Status

13. Write a R program to visualize student marks of various subjects using Bar-chart and Scatter plot

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 419	Information Retrieval	SE	3	0	0	3

UNIT I

Introduction to information retrieval, IR problem, IR system, The Web, Search interface, Visualizing search interface, Inverted index and boolean queries, Tokenization, Stemming, Stop words, Phrases, Phrases queries, Index construction, Index compression, k-gram indexes

UNIT II

Retrieval models: Boolean, Vector space model, TF-IDF, The cosine measure, Document length normalization, Probabilistic models, Binary Independence Model, Okapi, Language modeling,
Evaluating IR system: User happiness, Precision, Recall, F-measure, E-measure, Normalized recall, Evaluation problems

UNIT III

Relevance feedback and Query expansion: Explicit relevance feedback, Explicit relevance feedback through clicks, Implicit feedback through local analysis, Implicit feedback through global analysis
Document format, Markup language, Text properties, Document processing, Document organization, Text compression, Query languages, Query properties

UNIT IV

Text/Document classification, Clustering and LSI: Introduction to classification, Naive Bayes models, Rocchio classification, k-Nearest Neighbors, Support vector machine classifiers, Decision trees, Bagging, Boosting, Choosing right classifier

Introduction to clustering, Evaluation of clustering, k-means clustering, Hierarchical agglomerative clustering, Divisive clustering; Low-rank approximations, Latent semantic indexing

UNIT V

Web IR: Hypertext, Web crawling, Indexes, Search engines, Ranking, Link analysis, Page Rank, HITS

TEXTBOOKS & REFERENCES

1. Modern Information Retrieval: The Concepts and Technology Behind Search, by Ricardo Baeza-Yates and Berthier Ribeiro-Neto, Second Edition (Pearson Education India, 2010)
2. Introduction to Information Retrieval, by C. Manning, P. Raghavan, and H. Schütze (Cambridge University Press, 2008)
3. Mining the Web, by S. Chakrabarti (Morgan-Kaufmann, 2002)
4. Natural Language Processing And Information Retrieval, by Tanveer Siddiqui and U. S. Tiwary, First Edition (Oxford University Press, 2008)

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 419 L	Information Retrieval Lab	SE	0	0	2	1

LIST OF PRACTICAL EXPERIMENTS

1. Tokenization, Stemming, Stop words removal, Inverted index construction - Token sequence, Sort, Dictionary & Postings, Implementation of Boolean queries.
2. Ranked retrieval - Implementation of TF-IDF, Vector space model, Cosine similarity.
3. Ranked retrieval - Implementation of Binary Independence Model, Okapi BM25.
4. Implementation of Text/Document classification algorithms - Naive Bayes models, Rocchio classification, k-Nearest Neighbors, Support vector machine classifiers, Decision trees, Bagging, Boosting.
5. Implementation of Text/Document clustering algorithms - k-means clustering, Hierarchical agglomerative clustering, Divisive clustering.
6. Implementation of Low-rank approximations, Latent semantic indexing.
7. Sort-based index construction.
8. Implementation of External memory indexing - BSBI, SPIMI.
9. Implementations of Dynamic indexing - Logarithmic merge.
10. Dictionary compression - Implementation of Blocking, Posting Compression - Implementation of Gamma codes.
11. Development of a Web Crawler and a small-scale web search engine - Ranking, PageRank, HITS.

Distributed and Cloud Computing

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 316	Distributed Systems	SE	3	0	0	3

UNIT I: INTRODUCTION AND ARCHITECTURES

Introduction: Definition of a distributed system, Goals, types of distributed systems Architectures: Architecture styles, System architectures, Architectures versus middleware, Self-management in distributed systems.

UNIT II: PROCESSES AND COMMUNICATION

Processes: Threads, Virtualization, Clients, Servers, Code Migration. Communication: Fundamentals, Remote Procedure Call, Message and Stream oriented communication, Multicast communication.

UNIT III: NAMING AND SYNCHRONIZATION

Naming: Flat naming, Structured naming, Attribute-based naming. Synchronization: Clock synchronization, Logical clocks, Mutual exclusion, Election algorithms.

UNIT IV: CONSISTENCY AND REPLICATION

Replication as Scaling Technique, Data-Centric Consistency Models: Continuous Consistency, Data-Centric Consistency Models: Consistent Ordering of Operations, Data-Centric Consistency Models: Consistent Ordering Of Operations, Replica-Server Placement, Replica-Server Placement, Content Distribution, Continuous Consistency, Primary-Based Protocols, Replicated-Write Protocols, Cache-Coherence Protocols.

UNIT V: FAULT TOLERANCE AND SECUIRTY

Fault tolerance: Introduction, Process Resilience, Reliable client server communication, Reliable group communication, Distributed Commit, Recovery. Security: Secure channels, Access control, Security Management.

TEXTBOOKS/REFERENCES

1. Andrew S. Tanenbaul, Maarten Van Steen, Distributed Systems, Principles and Paradigms, Pearson publications, 2nd edition.
2. Pradeep K Sinha, “Distributed Operating Systems: Concepts and Design”, Prentice Hall of India, 2007.
3. George Coulouris, Jean Dollimore and Tim Kindberg, “Distributed Systems Concepts and Design”, Fifth Edition, Pearson Education, 2012.
4. Liu M.L., “Distributed Computing, Principles and Applications”, Pearson Education, 2004.

\

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 316 L	Distributed Systems Lab	SE	0	0	2	1

LIST OF PRACTICAL EXPERIMENTS

1. Experiment-1: Implement concurrent echo client-server application.
2. Experiment -2: Implement concurrent day-time client-server application.
3. Experiment-3: Configure following options on server socket and tests them: SO_KEEPALIVE, SO_LINGER, SO_SNDBUF, SO_RCVBUF, TCP_NODELAY
4. Experiment -4: Simulate the functioning of Lamport‟s Logical Clock in C.
5. Experiment -5: Simulate the Distributed Mutual Exclusion in C.
6. Experiment -6: Implement Java RMI‟ mechanism for accessing methods of remote systems.
7. Experiment -7: Simulate Balanced Sliding Window Protocol in C.
8. Experiment -8: Incrementing a counter in shared memory.
9. Experiment -9: Create CORBA based server-client application.
10. Experiment -10: Design XML Schema and XML instance document.
11. Experiment -11: SOAP based: Implement Arithmetic Service that implements add and subtract operations /Java based: Implement Trigonometric Service that implements sin, and cos operations.

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 318	Cloud Computing	SE	3	0	0	3

UNIT I

Distributed system models: Scalable computing over the internet, Technologies for network-based systems, System models and software environments for distributed and cloud computing, performance, security and Energy Efficiency Computer clusters for Scalable parallel computing: Clustering for Massive parallelism, Computer clusters and MPP Architectures, Design principles of computer clusters, Cluster job and resource management.

UNIT II

Virtual Machines and Virtualization of Data Centres: Implementation levels of virtualization, Virtualization structures, tools and mechanisms, Virtualization of CPU, Memory and I/O devices, Virtual clusters and resource management, Virtualization for Data center automation.

UNIT III: NAMING AND SYNCHRONIZATION

Cloud computing and service models, Data center design and interconnection networks, Architectural design of Compute and storage clouds, Public cloud platforms, Inter-cloud resource management, Cloud security and trust management.

UNIT IV: CONSISTENCY AND REPLICATION

Services and service-oriented architecture, Message oriented middleware, Portals and science gateways, Discovery, Registries, Meta data and databases. Workflow in service-oriented architectures,

UNIT V: FAULT TOLERANCE AND SECURITY

Features of cloud and Grid platforms, Parallel and distributed programming paradigms, Programming support for Google application engine, Programming on Amazon AWS and Microsoft Azure, Emerging cloud software environments.

TEXTBOOKS

1. Cloud Computing, Theory and Practice, Dan C Marinescu, MK Elsevier.
2. Cloud Computing: Principles and Paradigms, Rajkumar Buyya, James Broberg, Andrzej M. Goscinski, Wiley.

REFERENCES

1. Distributed and Cloud Computing. Kal Hwang. Geoffeiy C. Fox. Jack J. Dongarra. Elsevier. 2012.
2. Cloud computing, Black book. Deven Shah, Kailash Jayaswal, Donald J. Houde, Jagannath Kallakurchi.
3. Cloud Computing: Concepts, Technology & Architecture (The Prentice Hall Service Technology Series from Thomas Erl) 1st Edition, Thomas Erl (Author), Ricardo Puttini , Zaigham Mahmood.

Course Code	Course Name	Course Category	Credits

			L	T	P	C
CSE 318 L	Cloud Computing Lab	SE	0	0	2	1

LIST OF PRACTICAL EXPERIMENTS

1. Basics of Virtualization: VMM, Example of VMM (virtualbox), Cretaion of a VM, Networking and communication between VMs.
2. Introduction to CloudSim: Installation and Execution, Cloud Datacenter, Network Topology.
3. Simulation of a Cloud Framework: Creating a DC, Creation of Tasks, Creation of VMs, Defining task and VM characteristics, execution of tasks on VMs.
4. Scalable and dynamic Cloud systems: Creation of scalable cloud entities, creation of dynamic entities.
5. Resource Allocation in Cloud Datacenter: Experimenting and understanding various resource allocation policies, Changing the resource allocation policy, effects of resource allocation policies.
6. Power Management in Cloud Datacenters: Creation of a power datacenter, understanding various power saving techniques.
7. Understanding Commercial Cloud Frameworks: Amazon AWS, Elastic Cloud, Amazon Load Balancer, creating VMs, Allocation of Resources.

	Course Name	**Course Category**	**Credits**

Course Code			L	T	P	C
CSE 416	Cloud Data Management	SE	3	0	0	3

UNIT I: DISTRIBUTED FILE SYSTEM

Architecture: Client-Server Architectures, Cluster-Based Distributed File Systems, Symmetric Architectures, Processes, Communication, Naming, Synchronization, Consistency and replication, Fault tolerance, Security

UNIT II: DISTRIBUTED DATA MANAGEMENT

Distributed systems, peer to peer systems, database systems, Overview of key value stores and examples, Design choices and their implementations.

Transactions on co-located data: Data ownership, Transaction execution, Data storage, Replication, A survey of the systems.

UNIT III: CLOUD DATA MANAGEMENT

Database like functionality in cloud storage, Transactional support for geo-replicated data, Incremental update processing using distributed transaction, Scalable distributed synchronization using mini transactions. Multi-tenant database systems: Multitenancy models, database elasticity in the cloud, Autonomic control for data base workloads in the cloud.

UNIT IV: AZURE DATABASE SERVICE PLATFORM

Understanding the Service, Designing SQL Database, Migrating an Existing Database, Using SQL Database, Scaling SQL Database, Governing SQL Database. MySQL and PostgreSQL

UNIT V: SQL SERVER 2017

Hybrid cloud features, migrate databases to Azure IaaS, Run SQL Server on Microsoft Azure Virtual Machines, Considerations on High Availability and Disaster Recovery Options with SQL Server on Hybrid Cloud and Azure IaaS, Working with NoSQL Alternatives.

TEXTBOOKS

1. Data management in the cloud: challenges and opportunities: Divyakant Agrawal, Sudipto das, Amr EI Abbadi, 2013.
2. Cloud data design, Orchestration and Management using Microsoft Azure, Francesco Diaz Roberto Freato, Apress, Springer publications, 2018.

REFERENCES

1. Andrew S. Tanenbaul, Maarten Van Steen, Distributed Systems, Principles and Paradigms, Pearson publications, 2nd edition.
2. Cloud data design, Orchestration and Management using Microsoft Azure, Francesco Diaz Roberto Freato, Apress, Springer publications, 2018.
3. Cloud database development and Management, Lee chao, CRC Press, Taylor and Francis group. 2014.
4. Cloud data management, Liang Zhao, Sherif Sakr, Anna Liu, Athman Bouguettaya, Springer publications, 2014.

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 416 L	Cloud Data Management Lab	SE	0	0	2	1

LIST OF PRACTICALS EXPERIMENTS

1. **Week 1: Designing SQL Database in Azure**
   Using Database Projects (or other similar tools) to create and manage the development of the database.
   Connection modes:
   • outside Azure
   • inside Azure
2. **Week 2: SQL Database Options**
   Aanalyze the various SQL database options: Single Database with a Single Schema, Single Database with Different Schemas and other
3. **Week 3: Indexes**
   Index creation, Index evaluation, Index management for a table
   Automatic Tuning
4. **Week 4: Migration**
   Migrate an existing SQL Server database to Azure SQL Database.
   a. Preparing the Database
   b. Moving the Database
   c. Exporting the DB
5. **Week 5: Scaling**
   Dynamically scale database resources with minimal downtime
6. **Week 6: Governing SQL Database**
   Value-added services of SQL Database
   a. Authentication
   b. Firewall
   c. Encryption
7. **Week 7: Encryption**

Apply different database encryption methods to a database.
   a. Transparent Data Encryption
   b. Always Encrypted
   c. Dynamic Data Masking
   Explore different backup and monitoring options of Azure Databases

8. **Week 8: SQL Server 2017**
   Connect to SQL Server instance
   a. Create a database
   b. Create tables under the database
9. **Week 9: Azure Storage**
   Create a Storage Account in Azure portal
   Add a managed data disk to a SQL Server virtual machine
10. **Week 10: Backup on Azure**
    Create SQL Server Backup to URL.
11. **Week 11: Backup on Azure**
    Create a SQL Server Managed Backup to Microsoft Azure.

Take snapshots of data and log files that are placed into Azure Storage using File-Snapshots Backups.

12. **Week 12: Restore**
    Create a SQL Server Managed Backup to Microsoft Azure feature.
    Access backup data from Azure Storage taken using SQL Server Managed Backup and Restore it.
13. **Week 13:**
    Use Azure Storage to host SQL Server Database Files and Use Azure Snapshots
14. **Week 14:**
    Migrate a Database Using the Data-Tier Application Framework
15. **Week 15:**
    Use Azure Storage to host SQL Server Database Files and Use Azure Snapshots
    Explore the High Availability and Disaster Recovery Options with SQL Server On Hybrid Cloud and Azure IaaS

TEXTBOOKS/REFERENCES

1. Cloud data design, Orchestration and Management using Microsoft Azure, Francesco Diaz Roberto Freato, Apress, Springer publications, 2018.
2. "Design a relational database in Azure SQL Database using SSMS", https://docs.microsoft.com/en-us/azure/sql-database/sql-database-design-first-database
3. "Data Migration Assistant, https://www.microsoft.com/en-us/download/details.aspx?id=53595
4. "Dynamically scale database resources with minimal downtime", https://docs.microsoft.com/en-us/azure/azure-sql/database/scale-resources

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 418	Service Oriented Computing	SE	3	0	0	3

UNIT I: WEB SERVICE FUNDAMENTALS

Introduction to Web Services - fundamental of web services, basic operational model of web services, Business motivations for web services, B2B, B2C, Technical motivations, basic steps of implementing web services, benefits and challenges of using web services, tools and technologies enabling web services, Web services Architecture and its characteristics, web services communication models, core building blocks of web services, web services technology stack. Orchestration, Choreography. Service layer Abstraction - Application Service Layer, Business Service Layer, Orchestration Service Layer.

UNIT II: SERVICE ORIENTED ARCHITECTURE

Service–oriented Architecture (SOA), implementation view, logical view, process view, deployment view, composition of web services, from application server to peer to peer, life in the runtime. Characteristics of SOA, Comparing SOA to client-server and distributed internet architectures, Anatomy of SOA, How components in an SOA interrelate. Fundamentals of SOAP-SOAP Message Structure, SOAP encoding, Encoding of different data types, SOAP communication and messaging, SOAP message exchange models, limitations of SOAP. REST Protocol, SOAP vs REST.

UNIT III: SERVICE ORIENTED PLATFORMS

WSDL, Anatomy of WSDL, manipulating WSDL, web service policy, UDDI, Anatomy of UDDI, UDDI- UDDI registries, uses of UDDI Registry, UDDI data structures, Programming with UDDI, Publishing, searching and deleting information in a UDDI Registry, Publishing API, limitations of UDDI, Discovering Web Services, service discovery mechanisms, role of service discovery in a SOA, Service Selection. SOA support in J2EE: Java API for XML based web services (JAX-WS), Java architecture for XML binding (JAXB), Java API for XML Registries (JAXR), Java API for XML based RPC (JAXRPC), Web Services Interoperability Technologies (WSIT). SOA support in .NET: Common Language Runtime, ASP.NET web forms, ASP.NET web services, Web Services Enhancements (WSE).

UNIT IV: APPLICATION DEVELOPMENT USING OPEN STACK

Understanding Open stack eco system: Open stack Heat, Open stack Database As A Service: Trove, Designate: Dns As A Service, Magnum, Murano: Application As A Service, Ceilometer: Telemetry As A Service, Application development and deployment in Open stack: Building applications from the scratch, converting legacy applications into Open stack applications. Event Driven Programs with Cloud.

UNIT V: MONITORING AND METERING

Monitoring and metering, Updating and patching. Kubernetes: Concepts, Cluster Architecture, Containers and Dockers, Workloads, Services, Load Balancing, and Networking, Policies, Scheduling and Eviction, Cluster Administration. Apigee Edge, API development lifecycle.

TEXTBOOKS

1. Service-Oriented Architecture: Concepts, Technology, and Design By Thomas Erl, Pearson Education India.

2. OpenStack Cloud Application Development by Scott Adkins, John Belamaric, Vincent Giersch, Denys Makogon, Jason E. Robinson, Wrox.
3. Mastering kubernetes: Sayfan, Gigi, Packt Publishing Ltd.

REFERENCES

1. Service Oriented Computing: Semantics, Processes, Agents: Munindar Singh & Michael Huhns, Wiley Publication.
2. Enterprise SOA Designing IT for Business Innovation: Dan Woods and Thomas Mattern , O'REILLY.
3. Service-oriented Architecture for Enterprise Applications: Shankar Kambhampaty, John Wiley & Sons.
4. SOA using Java™ Web Services: Mark D Hansen, Prentice Hall Publication.

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 418 L	Service Oriented Computing Lab	SE	0	0	2	1

LIST OF PRACTICAL EXPERIMENTS

Develop Java Based Program using JAXP or XML API in reading XML file for Students Information and Display HTML Table.

1. Develop Java Based web Service using REST and SOAP Based web service in Netbeans for University Course List and Search Course based Course Title and Course ID.
2. Create web calculator service in .NET Beans and create Java client to consume this web service.
3. Develop same web service using JX-WS.
4. Create web calculator service in .NET and Create java client to consume web service. developed using Apache AXIS.
5. Using WS –GEN and WS-Import develop the java web service & call it by Java Client.
6. Open stack Heat.
7. Opens tack Database As A Service: Trove.
8. Designate: DNS As A Service.
9. Magnum.
10. Murano: Application As A Service.
11. Building applications from the scratch.
12. converting legacy applications into Open stack applications.
13. Kubernetes: Containers and Dockers.
14. Kubernetes: Load Balancing, Scheduling.

Internet of Things (IoT) Stream

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSES 337	Cryptography	SE	3	0	0	3

UNIT I

History and overview of cryptography, Classical Encryption Techniques: Symmetric Cipher Model, Substitution Techniques, Transposition Techniques, Rotor Machines, And Steganography

UNIT II

Stream Ciphers and Block Ciphers, Attacks on block ciphers, Block Cipher Principles, The Data Encryption Standard (DES), Block Cipher Design Principles, Group, Rings, Field, Polynomial Arithmetic, The Euclidean Algorithm, Finite Fields of the Form GF(2^n)

UNIT III

Advanced Encryption Standard (AES), Stream Ciphers, RC4, The Chinese Remainder Theorem, Public Key Cryptography and RSA Algorithm, Diffie-Hellman Key Exchange, Elliptic Curve Cryptography.

UNIT IV

Cryptographic Hash Functions: Applications of Cryptographic Hash Functions, Two Simple Hash Functions, Requirements and Security, Secure Hash Algorithm (SHA), SHA-3.

UNIT V

Introduction to Block Chain, Bitcoin basics, Smart Contracts, Blockchain development platforms and APIs, Blockchain Ecosystems, Ethereum, Distributed Consensus, Blockchain Applications

TEXTBOOKS/REFERENCES

1) Stallings, William. Cryptography and network security, Principle and Practice. Pearson Education India, 2017.
2) R. Stinson Cryptography, Theory and Practice (Fourth Edition Edition)
3) Handbook of Applied Cryptography by A. Menezes, P. Van Oorschot, S. Vanstone.
4) Melanie Swan, Blockchain, Blueprint for a new Economy, OReilly

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 337 L	Cryptography Lab	SE	0	0	2	1

LIST OF PRACTICAL EXPERIMENTS

1. Write a program take text file as an input and print word, character count and ascii value of each characters as output. (Hint: Use open(), read() and split()).
2. Write a encryption program: Input: computerscienceengineeringsrmuniversity Output: gsqtyxivwgmirgiirkmriivmrkwvqyrmzivwmxc Hint: key =4 (play with ascii value).
3. Raju send an encrypted message (cipher text) "PHHW PH DIWHU WKH WRJD SDUWB" to Rani. Can you build decryption process and find out what is the message (plain text) send to Rani? Hint: try all keys.
4. Raju send encrypted message "ZICVTWQNGKZEIIGASXSTSLVVWLA" to Rani. Can you build decryption process and find out what is the message send to Rani. Hint: try all keys for each character.
5. Kohli have plain text "wewishtoreplaceplayer". Can you build encryption process and find out what is the cipher text he needs send to BCCI. Help him out by using monoalphabatic cipher. Hint: use any one-to-one mapping between alphabets.
   One to one mapping

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	N	D	R	E	W	I	C	K	S	O	H	T	B	F	G	J	L	M	P	Q	U	V	X	Y	Z

6. Kohli sent encrypted message (Cipher text) "SEEMSEAOMEDSAMHL" to Anushka. Can you build decryption process and find out what is the message (plain text) send to Anushka. Hint: use above one to one mapping between alphabets.
7. Raju want to build encrypted and decryption algorithms of Playfair Cipher. Help him to build a key matrix using the key "srmapuniversity"
8. By using key matrix Raju want to send message "we are discovered save yourself" to Rani. Can you build encryption process and find out what is the cipher text message send to Rani by using palyfaircipher.

s	r	m	a	p
u	n	i/j	v	e
t	y	b	c	d
f	g	h	k	l
o	q	w	x	z

9. By using key "CBDE" Raju would like send message (plain text)"HELLO WORLD" to Rani. Can you build encryption process and find out what is the encrypted message (cipher text) to Raju by using Hill Cipher.Also Can you build decryption process and find out what is the decrypted message (plain text) of cipher text "SLHZYATGZT" by using Hill Cipher.
10. Implementation of Encryption and Decryption of Vigenère Cipher
    keyword *deceptive*
    key: deceptivedeceptivedeceptive
    plaintext: wearediscoveredsaveyourself
    ciphertext: ZICVTWQNGRZGVTWAVZHCQYGLMGJ
11. Implement the Encryption and Decryption of Row Transposition.

Key: 4 3 1 2 5 6 7
Plaintext: a t t a c k p
o s t p o n e
d u n t i l t
w o a m x y z
Ciphertext: TTNAAPTMTSUOAODWCOIXKNLYPETZ

12. Implement the Euclidean Algorithm for integers and polynomials.
13. Implement AES Key Expansion.
14. Implementation of AES encryption and decryption
15. Implementation of Simplified DES Encryption and decryption
16. Implementation of RC4
17. Implementation of RSA algorithm
18. Implementation of Diffie-Helman key exchanges
19. Implementation of elliptic-curve cryptography
20. Implementation of Hash functions

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 318	Cloud Computing	SE	3	0	0	3

UNIT I

Distributed system models: Scalable computing over the internet, Technologies for network-based systems, System models and software environments for distributed and cloud computing, performance, security and Energy Efficiency Computer clusters for Scalable parallel computing: Clustering for Massive parallelism, Computer clusters and MPP Architectures, Design principles of computer clusters, Cluster job and resource management.

UNIT II: PROCESSES AND COMMUNICATION

Virtual Machines and Virtualization of Data Centres: Implementation levels of virtualization, Virtualization structures, tools and mechanisms, Virtualization of CPU, Memory and I/O devices, Virtual clusters and resource management, Virtualization for Data center automation.

UNIT III: NAMING AND SYNCHRONIZATION

Cloud computing and service models, Data center design and interconnection networks, Architectural design of Compute and storage clouds, Public cloud platforms, Inter-cloud resource management, Cloud security and trust management.

UNIT IV: CONSISTENCY AND REPLICATION

Services and service-oriented architecture, Message oriented middleware, Portals and science gateways, Discovery, Registries, Meta data and databases. Workflow in service-oriented architectures.

UNIT V: FAULT TOLERANCE AND SECURITY

Features of cloud and Grid platforms, Parallel and distributed programming paradigms, Programming support for Google application engine, Programming on Amazon AWS and Microsoft Azure, Emerging cloud software environments.

TEXTBOOKS

1. Cloud Computing, Theory and Practice, Dan C Marinescu, MK Elsevier
2. Cloud Computing: Principles and Paradigms, Rajkumar Buyya, James Broberg, Andrzej M. Goscinski, Wiley.

REFERENCES

1. Distributed and Cloud Computing. Kal Hwang. Geoffeiy C. Fox. Jack J. Dongarra. Elsevier. 2012.
2. Cloud computing, Black book. Deven Shah, Kailash Jayaswal, Donald J. Houde, Jagannath Kallakurchi.
3. Cloud Computing: Concepts, Technology & Architecture (The Prentice Hall Service Technology Series from Thomas Erl) 1st Edition, Thomas Erl (Author), Ricardo Puttini , Zaigham Mahmood.

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 318 L	Cloud Computing Lab	SE	0	0	2	1

LIST OF PRACTICAL EXPERIMENTS

1. Basics of Virtualization: VMM, Example of VMM (virtual box), Creation of a VM, Networking and communication between VMs.
2. Introduction to Cloud Sim: Installation and Execution, Cloud Data centre, Network Topology,
3. Simulation of a Cloud Framework: Creating a DC, Creation of Tasks, Creation of VMs, Defining task and VM characteristics, execution of tasks on VMs.
4. Scalable and dynamic Cloud systems: Creation of scalable cloud entities, creation of dynamic entities.
5. Resource Allocation in Cloud Data centre: Experimenting and understanding various resource allocation policies, Changing the resource allocation policy, effects of resource allocation policies.
6. Power Management in Cloud Data centres: Creation of a power data centre, understanding various power saving techniques.
7. Understanding Commercial Cloud Frameworks: Amazon AWS, Elastic Cloud, Amazon Load Balancer, creating VMs, Allocation of Resources.

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 317	Embedded Systems	SE	3	0	0	3

UNIT I: INTRODUCTION TO EMBEDDED SYSTEMS

Introduction, characteristics of embedding computing applications, concept of real time systems, designing of hardware and software components, challenges in embedded system design, Safety and Security of an Embedded System, Performance of Embedded Systems.

UNIT II: INSTRUCTION SET OF PROCESSORS

Overview of various features of Computer Architecture, Instruction-set of ARM family of processors, Instruction-set of PIC family of Processors, Digital Signal Processor, Instruction set of TI C55X DSP. Programmed I/O, Interrupts (supported by Arm, PIC , TI C55x family of processors), Supervisor mode, exceptions, traps, co-processors, memory system, CPU power management.

UNIT III: INPUT-OUTPUT SUB-SYSTEM

I/O sub-system: busy-wait I/O, DMA, interrupt driven I/O, co-processors and hardware accelerators, Timers and counters, watchdog timers, interrupt controllers, DMA controllers, A/D and D/A converters. Component interfacing, interfacing protocols, Firewire, USB, IrDA. Sensors and Actuators.

UNIT IV: PROGRAM DESIGN AND ANALYSIS

State machine, circular buffer, stream-oriented programming, data flow graph (DFG), control flow graph (CFG), Compilation techniques, performance analysis, performance optimization, power analysis and power optimization, program validation and testing.

UNIT V: OPERATING SYSTEMS

Basic features of an operating system, Kernel features, polled loops system, co-routines, interrupt-driven system, multi-rate system, processes and threads, context switching, scheduling, task assignment, inter-process communication, Real-time Memory Management: Process stack management, dynamic allocation, synchronous and asynchronous I/O, Interrupt handling, device drivers, example real-time OS: VxWorks, RT-Linux, PSOS.

TEXTBOOKS

1. Wolf, Marilyn. Computers as components: principles of embedded computing system design. Elsevier, 2017 (4th Ed.).
2. Marwedel, Peter. Embedded System Design: Embedded Systems Foundations of Cyber-Physical Systems, and the Internet of Things. Springer, 2017. (3rd Ed.)

REFERENCES

1. Manish Patel, The 8051 Microcontroller based Embedded System, McGraw Hill 2014 (1st edn.).
2. Mall, Rajib. Real-time systems: theory and practice. Pearson Education India, 2009. (1st edn.).

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 317 L	Embedded Systems Lab	SE	0	0	2	1

LIST OF PRACTICAL EXPERIMENTS

1. Introduction to Software tool (preferably kiel MDK Microcontroller Development Kit) used in the lab. (2 hrs)
2. Interfacing of 8-bit ADC 0809 with 8051 Microcontroller. (1 hour)
3. Interfacing of 8-bit DAC 0800 with 8051 Microcontroller. (1 hour)
4. Implementation of Serial Communication by using 8051 serial ports. (1 hour)
5. Interfacing of individual LEDs and program them to blink after a fixed time interval. (1 hour)
6. Interfacing of 16*2 LCD panel with 8051 Microcontroller. (1 hour)
7. Interfacing of stepper motor with 8051 Microcontroller. (1 Hour)
8. A minor project is given to student to implement (7 hrs)

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 319	IoT Design Protocols	SE	3	0	0	3

UNIT I: OVERVIEW

IoT-An Architectural Overview– Building an architecture, Main design principles and needed capabilities, An IoT architecture outline, standards considerations. M2M and IoT Technology Fundamentals- Devices and gateways, Local and wide area networking, Data management, Business processes in IoT, Everything as a Service (XaaS), M2M and IoT Analytics, Knowledge Management.

UNIT II: REFERENCE ARCHITECTURE

IoT Architecture-State of the Art – Introduction, State of the art, Reference Model and architecture, IoT reference Model - IoT Reference Architecture- Introduction, Functional View, Information View, Deployment and Operational View, Other Relevant architectural views. Real-World Design Constraints- Introduction, Technical Design constraints-hardware is popular again, Data representation and visualization, Interaction and remote control.

UNIT III: IOT DATA LINK LAYER & NETWORK LAYER PROTOCOLS

PHY/MAC Layer (3GPP MTC, IEEE 802.11, IEEE 802.15), Wireless HART, Z-Wave, Bluetooth Low Energy, Zigbee Smart Energy, DASH7 - Network Layer-IPv4, IPv6, 6LoWPAN, 6TiSCH, ND, DHCP, ICMP, RPL, CORPL, CARP

UNIT IV: TRANSPORT & SESSION LAYER PROTOCOLS

Layer (TCP, MPTCP, UDP, DCCP, SCTP)-(TLS, DTLS) – Session Layer-HTTP, Co AP, XMPP, AMQP, MQTT.

UNIT V: SERVICE LAYER PROTOCOLS & SECURITY

Service Layer -oneM2M, ETSI M2M, OMA, BBF – Security in IoT Protocols – MAC 802.15.4, 6LoWPAN, RPL, Application Layer.

TEXTBOOKS/REFERENCES

1. Jan Holler, Vlasios Tsiatsis, Catherine Mulligan, Stefan Avesand, Stamatis Karnouskos,
2. David Boyle, “From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence”, 1st Edition, Academic Press, 2014.
3. Peter Waher, “Learning Internet of Things”, PACKT publishing, BIRMINGHAM –
4. MUMBAI
5. Bernd Scholz-Reiter, Florian Michahelles, “Architecting the Internet of Things”, ISBN 978-3-642-19156-5 e-ISBN 978-3-642-19157-2, Springer
6. Daniel Minoli, “Building the Internet of Things with IPv6 and MIPv6: The Evolving World of M2M Communications”, ISBN: 978-1-118-47347-4, Willy Publications Vijay Madisetti and Arshdeep Bahga, “Internet of Things (A Hands-on-Approach)”, 1st Edition, VPT, 2014.
7. http://www.cse.wustl.edu/~jain/cse570-15/ftp/iot_prot/index.html

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 319 L	IoT Design Protocols Lab	SE	0	0	2	1

LIST OF PRACTICAL EXPERIMENTS

Week1:

1. Study and Install IDE of Arduino and different types of Arduino Boards like Arduino uno, Arduino NG REV-C , Arduino NANO .

Week 2:

2. Write program using Arduino IDE for Blink LED.
Hardware Requirements:

1x Breadboard

1x Arduino Uno R3

1x RGB LED

1x 330Ω Resistor

2x Jumper Wires

Blinking the RGB LED:
With a simple modification of the breadboard, we could attach the LED to an output pin of the Arduino. Move the red jumper wire from the Arduino 5V connector to D13.

Week 3:

3. Develop a program using Arduino IDE and Arduino Board for RGB Led.
Hardware Requirements:
1x Breadboard

1x Arduino Uno R3

1x LED

1x 330Ω Resistor

2x Jumper Wires

Blinking the LED
With a simple modification of the breadboard, we could attach the LED to an output pin of the Arduino. Move the red jumper wire from the Arduino 5V connector to D13.

Week 4:

4. Study the temperature Sensors and write a program using Arduino IDE and Arduino Board for Temperature Sensor.

Weeks 5:

5. Study and Implement RFID, NFC using Arduino.
Hardware Requirements:
1 x Arduino UNO or 1 x Starter Kit for Raspberry Pi + Raspberry Pi
1 x Communication Shield
1 x RFID 13.56 MHz / NFC Module for Arduino and Raspberry Pi

1 x Mifare tag (card/keyring/sticker)
1 x PC

Weeks 6:

6. Write programs using Arduino IDE and Arduino Board for MQTT Protocol.

Weeks 7:

7. Write a program to Study and Configure Raspberry Pi.

Weeks 8

8. WAP for LED blink using Raspberry Pi.

Hardware Requirements:
1x Breadboard

1x Raspberry Pi

1x RGB LED

1x 330Ω Resistor

2x Jumper Wires

Weeks 9:

9. Study and Implement Zigbee Protocol using Raspberry Pi.

Week 10:

10. Study and implement 6LoWPAN Border Router Implementation for IoT Devices on Raspberry Pi.

Week 11:

11. Study and implement DTLS protocol for IoT devices using Raspberry Pi.

Week 12:

12. Study and implement CoAP protocol for IoT devices using Raspberry Pi.

Week 13:

13. Study and implement RPL protocol for IoT devices using Raspberry Pi.

Week 14

14. Study and implement MQTT protocol for IoT devices using Raspberry Pi.

Week 15:

15. Study and implement AMQP protocol for IoT devices using Raspberry Pi.

TECHNICAL ELECTIVES

TECHNICAL ELECTIVES

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 320	Web Programming	TE	3	0	0	3

UNIT I

Introduction to internet-Introduction to World Wide Web (WWW)-Web browsers-Web servers-Uniform Resource Locator (URL)- Introduction to Hyper Text Markup Language (HTML)-Standard HTML document structure-Text and Paragraph formatting- Lists in HTML-Handling of images in web pages-Hyperlinks- -Tables-Iframes in HTML-Forms in HTML-HTML Graphics-HTML Media

UNIT II

Introduction to Cascading Style Sheets (CSS)-CSS versions-The specification of CSS-Applying style to a document-Media types-Document structure and CSS inheritance-Selectors in CSS-Major themes of CSS-Style inclusion methods-CSS strings and keywords-CSS color values-Background attachment-border in CSS-Counter in CSS-Basics of Web fonts-CSS animations- CSS tool tips-CSS Image reflections-CSS grid container.

UNIT III

Overview of JavaScript-General syntactic characteristics of JavaScript-Primitives, Operations and Expressions-Control statements-Arrays-Functions-Constructors-Pattern matching using regular expressions-Error handling in JavaScript-Events and event handling-Document Object Model (DOM)-Dynamic documents with JavaScript-Positioning elements-moving elements-Changing colors and font-Dynamic content management-stacking elements-Locating mouse curser and Reacting to mouse click-Dragging and dropping elements

UNIT IV

Introduction to Hypertext Preprocessor (PHP)-General syntactic characteristics-Primitives, operations and expressions-Control statements-Arrays-Functions-Pattern matching in PHP-Form handling-Cookies and Session tracking-MySQL connectivity and various database operations with PHP

UNIT V

Introduction to Ajax-Ajax technology-Implementing Ajax-Applications-Ajax request-Ajax response-Ajax XML-Introduction to JSON-JSON syntax-JSON data types-JSON arrays-Introduction to Web APIs- Types of Web APIs-Examples of web APIs.

TEXTBOOKS

1. Thomas A. Powell, The Complete Reference HTML & CSS, Mc Graw Hill Publishers, Fifth Edition, 2017
2. Robert W. Sebesta, Programming the World Wide Web, Pearson Publishers, Eighth Edition, 2014.

REFERENCES

1. Richard Blum, PHP, MySQL & JavaScript All-in-one, Wiley, 2018

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 321	Human Computer Interaction	TE	3	0	0	3

UNIT I: FOUNDATIONS OF HCI

The Human: I/O channels – Memory – Reasoning and problem solving - The computer: Devices – Memory – Processing and networks - Interaction: Models – frameworks – Ergonomics – styles – elements – Interactivity- Paradigms.

UNIT II: DESIGN AND SOFTWARE PROCESS

Interactive design basics – Process – Scenarios – Navigation – Screen design – Iteration and prototyping - HCI in software process – Software life cycle – Usability engineering – Prototyping in practice – design rationale. Design rules – principles, standards, guidelines, rules. Evaluation Techniques – Universal Design.

UNIT III: MODELS AND THEORIES

Cognitive models –Socio-Organizational issues and stake holder requirements –Communication and collaboration Models-Hypertext, Multimedia and WWW.

UNIT IV: MOBILE HCI

Mobile Ecosystem: Platforms, Application frameworks- Types of Mobile Applications: Widgets, Applications, Games- Mobile Information Architecture, Mobile 2.0, Mobile Design: Elements of Mobile Design, Tools.

UNIT V: WEB INTERFACE DESIGN

Designing Web Interfaces – Drag and Drop, Direct Selection, Contextual Tools, Overlays, Inlays and Virtual Pages, Process Flow. Case Studies.

TEXTBOOKS

1. Alan Dix, Janet Finlay, Gregory Abowd, Russell Beale, “Human Computer Interaction”, Pearson Education.
2. Brian Fling, “Mobile Design and Development”, O’Reilly Media Inc. Bill Scott and Theresa Neil, “Designing Web Interfaces”, O’Reilly.

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 322	Advanced Computer Architecture	TE	3	0	0	3

UNIT I: INSTRUCTION LEVEL PARALLELISM

ILP – Concepts and challenges – Hardware and software approaches – Dynamic scheduling – Speculation - Compiler techniques for exposing ILP – Branch prediction.

UNIT II: MULTIPLE ISSUE PROCESSORS

VLIW & EPIC – Advanced compiler support – Hardware support for exposing parallelism– Hardware versus software speculation mechanisms – IA 64 and Itanium processors–Limits on ILP.

UNIT III: MULTIPROCESSORS AND THREAD LEVEL PARALLELISM

Symmetric and distributed shared memory architectures – Performance issues – Synchronization – Models of memory consistency – Introduction to Multithreading.

UNIT IV: MEMORY AND I/O

Cache performance – Reducing cache miss penalty and miss rate – Reducing hit time – Main memory and performance – Memory technology. Types of storage devices – Buses – RAID – Reliability, availability and dependability – I/O performance measures – Designing an I/O system.

UNIT V: MULTI-CORE ARCHITECTURES

Software and hardware multithreading – SMT and CMP architectures – Design issues – Case studies – Intel Multi-core architecture – SUN CMP architecture - heterogeneous multi-core processors – case study: IBM Cell Processor.

TEXTBOOKS

1. John L. Hennessey and David A. Patterson, “Computer architecture – A quantitative approach”, Morgan Kaufmann / Elsevier Publishers, 4th. edition, 2007.

REFERENCES

1. David E. Culler, Jaswinder Pal Singh, “Parallel computing architecture: A hardware/software approach”, Morgan Kaufmann /Elsevier Publishers, 1999.
2. Kai Hwang and Zhi.Wei Xu, “Scalable Parallel Computing”, Tata McGraw Hill, New Delhi, 200

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 323	Natural Language Processing	TE	3	0	0	3

UNIT I: INTRODUCTION

Natural Language Processing tasks in syntax, semantics, and pragmatics – Issues – Applications – The role of machine learning – Probability Basics –Information theory – Collocations -N-gram Language Models – Estimating parameters and smoothing – Evaluating language models.

UNIT II: WORD LEVEL AND SYNTACTIC ANALYSIS

Word Level Analysis: Regular Expressions-Finite-State Automata-Morphological Parsing-Spelling Error Detection and Correction-Words and Word Classes-Part-of Speech Tagging. Syntactic Analysis: Context-free Grammar-Constituency- Parsing-Probabilistic Parsing.

UNIT III: SEMANTIC ANALYSIS AND DISCOURSE PROCESSING

Semantic Analysis: Meaning Representation-Lexical Semantics- Ambiguity-Word Sense Disambiguation. Discourse Processing: Cohesion-Reference Resolution- Discourse Coherence and Structure.

UNIT IV: NATURAL LANGUAGE GENERATION AND MACHINE TRANSLATION

Natural Language Generation: Architecture of NLG Systems- Generation Tasks and Representations- Application of NLG. Machine Translation: Problems in Machine Translation-Characteristics of Indian Languages- Machine Translation Approaches-Translation involving Indian Languages.

UNIT V: INFORMATION RETRIEVAL AND LEXICAL RESOURCES

Information Retrieval: Design features of Information Retrieval Systems-Classical, Non-classical, Alternative Models of Information Retrieval – valuation Lexical Resources: WorldNet-Frame Net-Stemmers-POS Tagger- Research Corpora.

TEXTBOOKS

1. Daniel Jurafsky, James H. Martin, “Speech & language processing”, Pearson publications.
2. James Allen, Natural Language Understanding. The Benajmins/Cummings Publishing Company Inc. 1994. ISBN 0-8053-0334-0
3. Bird, Steven, Ewan Klein, and Edward Loper, Natural language processing with Python: Analyzing text with the natural language toolkit, O'Reilly Media, Inc, 2009.
4. Manning, Christopher, and Hinrich Schutze. Foundations of statistical natural language processing. MIT press, 1999.

REFERENCES

1. Pierre M. Nugues, “An Introduction to Language Processing with Perl and Prolog”, Springer.
2. Cover, T. M. and J. A. Thomas, Elements of Information Theory, Wiley, 1991. ISBN 0-471-06259-6.

3. Charniak, E.: Statistical Language Learning. The MIT Press. 1996. ISBN 0-262-53141-0.
4. Tom Mitchell, Machine Learning. McGraw Hill, 1997. ISBN 0070428077.

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 324	Computer Graphics	TE	3	0	0	3

UNIT I: INTRODUCTION

Application areas of Computer Graphics, overview of graphics systems, video-display devices, raster-scan systems, random scan systems, graphics monitors, and workstations and input devices

Output primitives: Points and lines, line drawing algorithms, mid-point circle and ellipse algorithms. Filled area primitives: Scan line polygon fill algorithm, boundary-fill, and flood-fill algorithms.

UNIT II: 2-D GEOMETRICAL TRANSFORMS

Translation, scaling, rotation, reflection and shear transformations, matrix representations and homogeneous coordinates, composite transforms, transformations between coordinate systems.

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

UNIT III: 3-D OBJECT REPRESENTATION

Polygon surfaces, quadric surfaces, spline representation, Hermite curve, Bezier curve and B-spline curves, Bezier and B-spline surfaces. Basic illumination models, polygon rendering methods.

3-D Geometric transformations: Translation, rotation, scaling, reflection and shear transformations, composite transformations, 3-D viewing: Viewing pipeline, viewing coordinates, view volume and general projection transforms and clipping.

UNIT IV: VISIBLE SURFACE DETECTION METHODS

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

UNIT V: COMPUTER ANIMATION

Design of animation sequence, general computer animation functions, raster animation, computer animation languages, key frame systems, motion specifications

TEXTBOOKS:

1. Computer Graphics with Virtual Reality System, Rajesh K. Maurya, Wiley Dreamtech.
2. Computer Graphics, D. Hearn and M.P. Baker (C Version), Pearson Education

REFERENCES:

1. Computer Graphics Principle and Practice, J.D. Foley, A.Dam, S.K. Feiner, Addison, Wesley
2. “Procedural elements for Computer Graphics”, David F Rogers, Tata Mc Graw hill, 2nd edition.
3. “Principles of Interactive Computer Graphics”, Neuman and Sproul, TMH.
4. Principles of Computer Graphics”, Shalini, Govil-Pai, Springer.

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 325	Advanced Data Structures and Algorithms	TE	3	0	0	3

UNIT I: ADVANCED DATA STRUCTURES

Strategies for choosing the appropriate data structures-Heaps, AVL Trees (Search, Insertion, Deletion, Red-Black Trees (Search, Insertion and Deletion), Splay Trees (Search, Insertion and Deletion), B-trees, B+ Trees (Search, Insertion and Deletion), Fibonacci heaps, Data Structures for Disjoint Sets, Augmented Data Structures.

UNIT II: GRAPHS & ALGORITHMS

Cut-sets, Connectivity and Separability, Planar Graphs, Isomorphism, Graph Coloring, Covering and Partitioning, Topological sort, Max flow: Ford Fulkerson algorithm, max flow – min cut, Dynamic Graphs, Few Algorithms for Dynamic Graphs, Union-Find Algorithms.

UNIT III: GEOMETRIC ALGORITHMS

Point location, Convex hulls and Voronoi diagrams, Arrangements, graph connectivity, Network Flow and Matching: Flow Algorithms - Maximum Flow – Cuts - Maximum Bipartite Matching - Graph partitioning via multi-commodity flow, Karger'r Min Cut Algorithm, String matching and document processing algorithms.

UNIT IV: APPROXIMATION ALGORITHMS

Approximation algorithms for known NP hard problems - Analysis of Approximation Algorithms. Use of Linear programming and primal dual, Local search heuristics. Parallel algorithms: Basic techniques for sorting, searching, merging, list ranking in PRAMs and Interconnection.

UNIT V: RANDOMIZED ALGORITHMS

Introduction, Type of Randomized Algorithms, Min- Cut, 2-SAT, Game Theoretic Techniques, Random Walks. Online Algorithms: Introduction, Online Paging Problem, Adversary Models, k-server Problem

TEXTBOOKS

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, “Introduction to Algorithms”, Third Edition, The MIT Press, 2009.

REFERENCES

1. Sahni, Sartaj, Data Structures, Algorithms and Applications in C++, MIT Press (2005)
2. Roger Sedgewick and Kevin Wayne, Algorithms, Addison-Wesley Professional 2011.
3. Allan Borodin and Ran El-Yaniv: Online Computation and Competitive Analysis, Cambridge University Press, 2005.
4. Sanjoy Dasgupta, Christos Papadimitriou and Umesh Vazirani, “Algorithms”, Tata McGraw-Hill, 2009.
5. RK Ahuja, TL Magnanti and JB Orlin, “Network flows: Theory, Algorithms, and Applications”, Prentice Hall Englewood Cliffs, NJ 1993.

6. Rajeev Motwani, Prabhakar Raghavan: Randomized Algorithms, Cambridge University Press, 1995.
7. Jiri Matousek and Bernd Gärtner: Understanding and Using Linear Programming, 2006.

Course Code	Course Name	Course Category	Credits			
			L	T	P	C

CSE 326	Distributed Operating Systems	TE	3	0	0	3

UNIT I: FUNDAMENTALS

What is distributed operating system, issues in designing distributed operating system, Computer networks: Lan, WAN technologies, communication protocols, internetworking, Message passing: Issues in IPC by message passing, synchronization, buffering group communication, case study.

UNIT II: REMOTE PROCEDURE CALLS

The RPC model, Implementing RPC, RPCs in heterogeneous environment, lightweight RPC, case study. Distributed shared memory: General architecture of DSM systems, Design and implementation issues of DSM, Consistency models, Replacement strategies, Advantages of DSM.

UNIT III: PROCESS MANAGEMENT

Introduction, Process migration, Threads. Synchronization: Clock synchronization, event ordering, Mutual exclusion, deadlock, Election Algorithms. Resource management: Global scheduling algorithm, Task assignment, Load sharing and balancing approaches.

UNIT IV: DISTRIBUTED FILE SYSTEM

Desirable features of a good DFS, file models, file accessing models, file sharing semantics, file caching schemes, file replication, fault tolerance, atomic transactions, Design principles, Case study: Google DFS and Hadoop DFS.

UNIT V: NAMING

Desirable features of a good naming system, system-oriented names, object locating mechanisms, human oriented names, name caches, naming and security. Security: potential attacks, cryptography, authentication, access control, digital signatures, design principles.

TEXTBOOKS/REFERENCES

1. Pradeep K Sinha, "Distributed Operating Systems: Concepts and Design", Prentice Hall of India, 2007.
2. Advanced Concepts in Operating Systems, Mukesh Singhal and Niranjan Shivratri, Mc Graw hill publications, 2017
3. Andrew S. Tanenbaul, Maarten Van Steen, Distributed Systems, Principles and Paradigms, Pearson publications, 2nd edition.

Course Code	Course Name	Course Category	Credits			
			L	T	P	C

CSE 420	Data and Web Mining	TE	3	0	0	3

UNIT I: INTRODUCTION TO DATA MINING

What is data mining? Related technologies - Machine Learning, DBMS, OLAP, Statistics. Data Mining Goals. Stages of the Data Mining Process, Data Mining Techniques, Knowledge Representation Methods. Data Warehouse and OLAP: Data Warehouse and DBMS, Multidimensional data model, OLAP operations .

UNIT II: DATA PRE-PROCESSING

Data cleaning. Data transformation, Data reduction. Data mining knowledge representation, Attribute-oriented analysis. Data mining algorithms: Association rules: Motivation and terminology, Basic idea: item sets, generating item sets and rules efficiently, Correlation analysis.

UNIT III: DATA MINING ALGORITHMS

Classification, Basic learning/mining tasks, inferring rudimentary rules: 1R algorithm, Decision trees, Covering rules. Data mining algorithms: Prediction, The prediction task, Statistical (Bayesian) classification, Bayesian networks, Instance-based methods (nearest neighbour), Linear models.

UNIT IV: WEB CRAWLING

Basic crawler algorithm, Focused crawlers, Topical crawlers, Web search: Web page pre-processing, Inverted index, HITS algorithm, Page ranking algorithm, Leadership algorithm.

UNIT V: SOCIAL NETWORK ANALYSIS

Co-citation and bibliographic coupling, Community discovery. Web usage mining: Recommender systems. Mining Twitter, Mining Face book, Mining Instagram.

TEXTBOOKS/REFERENCES

1. Han, J., Kamber, M., & Pei, J. (2011). Data mining: Concepts and techniques (3rd ed.). Morgan Kaufmann publications.
2. Introduction to Data Mining, Vipin kumar, Michael Steinbach, Pang-Ning Tan, Person publications,2016
3. Mining the Web, Soumen Chakrabarti, Elseier publications, 2002
4. Web Data Mining, Bing Liu, Second Edition, Springer publications, 2011.
5. Mining the Social Web, Mathew A. Russel, Mikhail Klassen, Third edition, Oreily publications, 2018.

Course Code	Course Name	Course Category	Credits

			L	T	P	C
CSE 421	Complexity Theory	TE	3	0	0	3

UNIT I: COMPUTABILITY

A recap of automata theory and the Church-Turing Thesis Computational models: Lambda calculus, Turing machine Decidability Reducibility. The PCP problem & Mapping reducibility The Recursion Theorem Definition of Information.

UNIT II: TIME COMPLEXITY

Measuring Complexity, Big-O and small-o notation, Analyzing algorithms. Complexity relationships among computational models The Class-P, Examples The Class-NP, Examples The P versus NP question NP-completeness The Cook-Levin Theorem Additional NP-completeness Problems.

UNIT III: SPACE COMPLEXITY

Space complexity. Savitch's Theorem and NL. NL-completeness and log-space reductions. From P-completeness to PSPACE-completeness. The Classes L and NL NL completeness, NL equals coNL.

UNIT IV: INTERACTABILITY

Hierarchy Theorems Relativization Circuit Complexity.

UNIT V: ADVANCED TOPICS IN COMPLEXITY THEORY

Approximation Algorithms Probabilistic Algorithms Alternation Interactive Proof Systems.

TEXTBOOKS

1. Introduction to the Theory of Computation - Michael Sipser (Primary Textbook)
2. Computational Complexity - Arora Barak (Reference)

Course Code	Course Name		Credits

		Course Category	L	T	P	C
CSE 422	Software Project Management	TE	3	0	0	3

UNIT I: SOFTWARE MANAGEMENT & ECONOMICS

SDLC -waterfall model Conventional Software Management Performance Evolution of Software Economics – Software economics Pragmatic software cost estimation Reducing software product size Improving software processes Improving team effectiveness Improving automation through software environment.

UNIT II: THE OLD AND THE NEW WAY OF PROJECT MANAGEMENT

The principles of conventional software engineering Principles of modern software management, Transitioning to an iterative process Basics of Software estimation – Effort and Cost estimation techniques COSMIC Full function points COCOMO-I COCOMO II A Parametric Productivity Model - Staffing Pattern.

UNIT III: SOFTWARE MANAGEMENT PROCESS FRAMEWORK

Life cycle phases: Engineering and production stages, inception, Elaboration, construction, transition phases. Artifacts of the process: The artifact sets, Management artifacts, Engineering artifacts, programmatic artifacts Model based software architectures: A Management perspective. Model based software architectures: Technical perspective Work Flows of the process: Software process workflows Iteration workflows Checkpoints of the process: Major milestones, Minor Milestones, Periodic status assessment.

UNIT IV: PROJECT ORGANIZATION AND PLANNING

Work breakdown structures Planning guidelines. The cost and schedule estimating process The iteration planning process Pragmatic planning Line-of-Business organizations Project organizations, Evolution of organizations Process automation - Automation building Blocks The project environment.

UNIT V: PROJECT CONTROL AND PROCESS INSTRUMENTATION

The Seven-Core metrics: Management indicators The Seven-Core metrics: Quality indicators Life-Cycle expectations, Pragmatic software metrics, Metrics automation Modern project profiles Next generation software economics Modern process transitions.

TEXBOOKS/REFERENCES

1. Walker Royce, “Software Project Management”, 1st Edition, Pearson Education, 2006.
2. Bob huges, Mike cotterell, Rajib Mall “Software Project Management”, 6th Edition, Tata McGraw Hill, 2017.
3. SA Kelkar, Software Project Management: A Concise Study, 3rd Edition, PHI, 2013.
4. Joel Henry, Software Project Management: A Real-World Guide to Success, Pearson Education, 2009.
5. Pankaj Jalote, Software Project Management in Practice, Pearson Education, 2015.
6. https://ocw.mit.edu/courses/engineering-systems-division/esd-36-system-project-management-fall-2012/
7. https://uit.stanford.edu/pmo/pm-life-cycle

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 423	Multimedia	TE	3	0	0	3

UNIT I: INTRODUCTION TO MULTIMEDIA

What is Multimedia, Multimedia and Hypermedia, Overview ofMu1timedia Software Tools Graphics and Multimedia Data Representations: Graphics Image Data Types, File Formats, and representation (image, video, and sound).

UNIT II: COLOUR IN IMAGE AND VIDEO

Color Science, Color' Models in Images, Color Models in Video, Fundamental Concepts in Video, Analog Video, Digital Video Basics of Digital Audio: Digitization of Sound, MIDI: Musical Instrument Digital Interface Quantization and Transmission of Audi.

UNIT III: LOSSLESS COMPRESSION ALGORITHMS

Basics of Infonnation Theory, Run-Length Coding, Variable-Length Coding, Dictionary-Based Coding, Arithmetic Coding, Lossless Image Compression Lossy Compression Algorithms: Distortion Measures, The Rate-Distortion Theory Quantization, Transform Coding, Wavelet-Based Coding, Embedded Zerotree of Wavelet Coefficients.

UNIT IV: **IMAGE COMPL'ESSION STANDARDS**

The JPEG Standard, The JPEG2000 Standard, The JPEG-LS Standard, Bilevel Image Compression Standards.

Basic Video Compression Techniques: Introduction to Video Compression, Video Compression Based on Motion Compensation, Search for Motion Vectors, H.261, H.263.

Basic Audio Compression Techniques: ADPCM in Speech Coding, G.726 ADPCM, Vocoders.

UNIT V: MPEG Video Coding I - MPEG-1 and 2

MPEG-1, MPEG-2 MPEG Video Coding 11- MPEG-4, 7, and Beyond: Overview ofMPEG-4, Object-Based Visual Coding in MPEG-4, Synthetic Object Coding in MPEG-4, MPEG-4 Part10/H.264, MPEG-7, H.265 MPEG Audio Compl'ession: MPEG Audio, Commercial Audio codes.

TEXTBOOKS

1. Fundamentals of Multimedia (FM), Ze-Nian Li, Mark S. Drew, in Prentice Hall, 2004 (Springer 2nd Edition, 2014 with additional author of Dr. Jiangchuan Liu).
2. Digital Multimedia by Chapman (DM), Nigel P./ Chapman, Jenny, in John Wiley & Sons Inc, 2000 (3rd Edition, 2009).

REFERENCES

1. Multimedia: Making It Work, 9 Edition by Vaughan, Tay in McGraw-Hill, 2014.
2. Multimedia: Computing, Communications and Applications by Ralf Steinmetz in Pearson Education, 2012.
3. Recent articles about multimedia (recommended at classes).

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 424	Deep Learning	TE	3	0	0	3

UNIT I: INTRODUCTION

Overview of machine learning, linear classifiers, loss functions.

Introduction to Tensor Flow: Computational Graph, Key highlights, creating a Graph, Regression example, Gradient Descent, Tensor Board, Modularity, Sharing Variables, Keras.

UNIT II: Activation Functions

Sigmoid, ReLU, Hyperbolic Fns, Softmax Perceptrons: What is a Perceptron, XOR Gate.

Artificial Neural Networks: Introduction, Perceptron Training Rule, Gradient Descent Rule, vanishing gradient problem and solution.

UNIT-III: Convolutional Neural Networks

Introduction to CNNs, Kernel filter, Principles behind CNNs, Multiple Filters, problem, and solution of under fitting and over fitting.

UNIT IV: Recurrent Neural Networks

Introduction to RNNs, Unfolded RNNs, Seq2Seq RNNs, LSTM, GRU, Encoder Decoder architectures.

UNIT V: Deep Learning applications

Image segmentation, Self-Driving Cars, News Aggregation and Fraud News Detection Natural Language Processing, Virtual Assistants, Entertainment, Visual Recognition Fraud Detection, Healthcare.

TEXTBOOKS

1. Goodfellow, I., Bengio, Y., and Courville, A., Deep Learning, MIT Press, 2016.
2. Josh Patterson, Adam Gibson, Deep Learning: A Practitioner's Approach, OReilly, 2017.
3. Gulli, Antonio, and Sujit Pal. Deep learning with Keras. Packt Publishing Ltd, 2017.
4. Buduma, Nikhil, and Nicholas Locascio. Fundamentals of deep learning: Designing next-generation machine intelligence algorithms. " O'Reilly Media, Inc.", 2017.

REFERENCES

1. Bishop, C., M., Pattern Recognition and Machine Learning, Springer, 2006.
2. Yegnanarayana, B., Artificial Neural Networks PHI Learning Pvt. Ltd, 2009.
3. Golub, G., H., and Van Loan, C. F., Matrix Computations, JHU Press,2013.
4. Satish Kumar, Neural Networks: A Classroom Approach, Tata McGraw-Hill Education, 2004.

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 425	Advanced Database Management Systems	TE	3	0	0	3

UNIT I

Overview of the DBMS Introduction to DBMS implementation using Megatron 2000 database system Data storage using main memory and hard disks Disk failures Recovery from disk crashes Representing data elements: Record, Representing block and record address Variable length data and records Record modifications.

UNIT II

Index structures: Indexes on sequential files Secondary indexes B-Trees Hash tables Multidimensional indexes: Hash and tree like structures for multidimensional data Bitmap indexes.

UNIT III

Query execution: Algebra for queries Introduction to Physical-Query-Plan Operators One-Pass Algorithms for Database Operations Nested-Loop Joins Two-Pass Algorithms Based on Sorting Two-Pass Algorithms Based on Hashing Index-Based Algorithms Buffer Management Algorithms Using More Than Two Passes Parallel Algorithms for Relational Operations.

UNIT IV

The query compiler: Parsing Algebraic Laws for Improving Query Plans from Parse Trees to Logical Query Plans Estimating the Cost of Operations Introduction to Cost-Based Plan Selection Choosing an Order for Joins Completing the Physical-Query-Plan Selection.

UNIT V

Concurrency control: Conflict-Serializability View serializability Enforcing Serializability by Locks Locking Systems with Several Lock Modes. An Architecture for a Locking Scheduler Concurrency control by timestamps and validation Transactions that Read Uncommitted Data Coping with system failures: Undo/Redo logging Protecting media failures

TEXTBOOKS

1. R. Ramakrishnan, J. Gehrke, Database Management Systems, McGraw Hill, 2004.
2. A. Silberschatz, H. Korth, S. Sudarshan, Database system concepts, 5/e, McGraw Hill, 2008.

REFERENCES

1. K. V. Iyer, Lecture notes available as PDF file for classroom use.

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 426	Fog Computing	TE	3	0	0	3

UNIT I: FOG COMPUTING

Limitation of Cloud computing, Differences between Cloud and Fog computing, what is Fog? Advantages of Fog computing, Business Models, Architecture of Fog computing, Opportunities and Challenges.

UNIT II: ADDRESSING THE CHALLENGES IN FOG RESOURCES

Introduction, Taxonomy and Characteristics, Resource Management Challenge, Optimisation challenges, Miscellaneous Challenges, IoT and Fog: Introduction. Programming paradigms for IoT+ Fog, Research challenges and Future Research Directions.

UNIT III: MANAGEMENT AND ORCHESTRATION OF NETWORK SLICES IN 5G, FOG, EDGE, AND CLOUDS

Introduction, Background, Network Slicing in 5G, Network Slicing in Software-Defined Clouds, Network Slicing Management in Edge and Fog, Future Research Directions: Middleware for Fog and Edge Computing: Design Issues, Introduction. Need for Fog and Edge Computing Middleware: Design Goals, State-of-the-Art Middleware Infrastructures, System Model, Clusters for Lightweight Edge Clouds, Architecture Management – Storage and Orchestration, IoT Integration, Security Management for Edge Cloud Architectures, Future Research Directions.

UNIT IV: DATA MANAGEMENT AND ANALYSIS IN FOG COMPUTING

Introduction, Background, Fog Data Management, Future Research and Direction Motivating Example: Smart Building, Predictive Analysis with Fog Torch, Survey of ML Techniques for Defending IoT Devices, Machine Learning in Fog Computing, Future Research Directions.

UNIT V: CASE STUDIES

Case Study 1: Introduction, Human Object Detection, Object Tracking, Lightweight Human Detection. Case Study 2: Introduction, Data-Driven Intelligent Transportation Systems, Mission-Critical Computing Requirements of Smart Transportation Applications, Fog Computing for Smart Transportation Applications, Case Study 3: Intelligent Traffic Lights Management (ITLM) System, Testing Perspectives.

TEXTBOOKS

1. Fog and Edge Computing, Rajkumar Buyya, Satish Narayana Srirama, Wiley Publications, 2019.
2. Fog computing in the Internet of Things: Springer publications, 2018

REFERENCES

1. Research papers from IEEE, ACM, Springer and Elsevier)

Course Code	Course Name	Course Category	Credits

			L	T	P	C
CSE 427	Parallel Algorithms	TE	3	0	0	3

UNIT I

Sequential model need of alternative model, parallel computational 8 models such as PRAM, LMCC, Hypercube, Cube Connected Cycle, Butterfly, Perfect Shuffle Computers, Tree model, Pyramid model, Fully Connected model, PRAM-CREW, EREW models, simulation of one model from another one.

UNIT II

Performance Measures of Parallel Algorithms, speed-up and 8 efficiency of PA, Cost- optimality, an example of illustrate Cost- optimal algorithms- such as summation, Min/Max on various models.

UNIT III

Parallel Sorting Networks, Parallel Merging Algorithms on on 8 CREW/EREW/MCC, Parallel Sorting Networks CREW/EREW/MCC/, linear array.

UNIT IV

Parallel Searching Algorithm, Kth element, Kth element in X+Y on 8 PRAM, Parallel Matrix Transportation and Multiplication Algorithm on PRAM, MCC, Vector-Matrix Multiplication, Solution of Linear Equation, Root finding.

UNIT V

Graph Algorithms - Connected Graphs, search and traversal, 8 Combinatorial Algorithms- Permutation, Combinations, Derangements.

TEXTBOOKS

1. M.J. Quinn, “Designing Efficient Algorithms for Parallel Computer”, Mc Graw Hill.
2. S.G. Akl, “Design and Analysis of Parallel Algorithms” 3. S.G. Akl,” Parallel Sorting Algorithm” by Academic Press

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 428	Web Services	TE	3	0	0	3

UNIT I

Introduction to Service Oriented Architecture-Goals of service oriented architecture- Introduction to services-The SOA Architectural Stack-Service Composition and Data Flow-Data-Flow Paradigms-Composition Techniques

UNIT II

Introduction to web services- History of webservices-Web services: communication stack-Simple Object Access Protocol (SOAP)-Web Services Description Language (WSDL)-WSDL Main Elements-Message Communication Model in SOAP/WSDL

UNIT III

Web Services: REST or Restful Services-REST Design Principles-Web API Design for RESTful Services-Data Services-Implementation of Data Services-XML Transformation and Query Techniques-Consuming data via direct data access to the sources

UNIT IV

Web Service Composition: Overview-Service Orchestration vs. Service Choreography-Benefits of Web Service Composition-Web Service Composition Environment-Web Service Composition: Control Flows-BPEL (Business Process Execution Language)-BPMN (Business Process Model and Notation)-Web Service Composition: Data Flows-Data-Flow Paradigms

UNIT V

Introduction to Service Component Architecture (SCA)-The SOA Integration Problem-Overview of SCA-High-level overview of the assembly model-Application of SCA to Use Case-SCA Runtime-Benefits of SCA

TEXTBOOKS

1. Paik, Hye-young, et al. Web Service Implementation and Composition Techniques. Vol. 256. Springer International Publishing, 2017.
2. Martin Kalin, Java Web Services: Up and Running, O'Reilly publishers, Second edition, 2013.

Course Code	Course Name	Course Category	Credits			
			L	T	P	C
CSE 429	Advances in Data Mining	TE	3	0	0	3

UNIT I

What is Data Mining, Compiling need of Data Mining, Business Data Mining, Data Mining Tools. Data Mining Process, CRISP-DM, Business Understanding, Data Understanding, Data Preparation, Modelling, Evaluation, Deployment. SEMMA, Steps in SEMMA Process, Comparison of CRISP & SEMMA, Handling Data.

UNIT II

Association Rules in Knowledge Discovery, Market-Basket Analysis, Mining Frequent Patterns, Associations, and Correlations, Apriori Algorithm, Pattern-Growth Approach for Mining Frequent Itemsets, Mining Frequent Itemsets using Vertical Data Format, Mining Closed and Max Patterns. Pattern Mining in Multilevel, Multidimensional Space, Constraint-Based Frequent Pattern Mining, Mining High-Dimensional Data and Colossal Patterns, Mining Compressed or Approximate Patterns.

UNIT III

Classification: Basic Concepts, Decision Tree Induction, Bayes Classification Methods: Bayes' Theorem, Na¨ıve Bayesian Classification, Rule-Based Classification. Model Evaluation and Selection, Techniques to Improve Classification Accuracy: Bagging, Boosting and AdaBoost, Random Forests, Improving Classification Accuracy of Class-Imbalanced Data. Other Classification Methods: Genetic Algorithms, Rough Set Approach, Fuzzy Set Approaches.

UNIT IV

Cluster Analysis, Partitioning Methods: k-Means: A Centroid-Based Technique, k-Medoids: A Representative Object-Based Technique. Hierarchical Methods: Agglomerative versus Divisive Hierarchical Clustering, Distance Measures in Algorithmic Methods, BIRCH: Multiphase Hierarchical Clustering Using Clustering, Feature Trees, Chameleon: Multiphase Hierarchical Clustering Using Dynamic Modelling, Probabilistic Hierarchical Clustering. Density-Based Methods, Grid-Based Methods.

UNIT V

Outliers and Outlier Analysis, Outlier Detection Methods: Supervised, Semi-Supervised, and Unsupervised Methods, Statistical Methods, Proximity-Based Methods, and Clustering-Based Methods, Mining Contextual and Collective Outliers, Outlier Detection in High-Dimensional Data. Mining Complex Data Types, Data Mining Applications, Social Impacts of Data Mining.

TEXTBOOKS

1. Data Mining Concepts and Techniques, Third Edition, by Jiawei Han, Micheline Kamber, and Jian Pei.
2. Olson DL, Delen D. Advanced data mining techniques. Springer Science & Business Media.

REFERENCES

1. Aggarwal CC. Data mining: the textbook. Springer. William
2. Machine Learning, 2nd edition, by Ethem Alpaydi

COURSE CODE	COURSE NAME	CORE/ ELECTIVE	CREDITS			
			L	T	P	C
CSE 327	Social Network Analysis	E	3	0	0	3

Unit – 1: Fundamentals of Network Science

Introduction

Networks in the real world: Social networks, Information networks, Technological networks, Biological networks

The large-scale structure of networks: Components, Shortest paths and small-world effect, Degree distributions, Power laws and scale-free networks, Six degrees of separation, Random graphs models of network formation.

Mathematics of networks: Networks and their representation, Types of networks: Weighted, directed and hypergraphs, The adjacency, Laplacian, and incidence matrices, Degree, paths, components, independent paths, connectivity, and cut sets.

UNIT -II: Centrality measures

Degree centrality, Closeness centrality, Homophily, Transitivity and Preferential attachment, Clustering coefficient and Assortative mixing, Eigenvector centrality, Katz centrality, Betweenness centrality, Page rank, Hubs and Authorities

UNIT – III: Community Detection in Social Networks

Detecting communities in social networks, Definition of community, Applications of community detection.

Algorithms for community detection: The Kernighan-Lin Algorithm, Agglomerative/Divisive Algorithms, Markov Clustering, Multi-level Graph Partitioning, Spectral Algorithms, Modularity Maximization, Other Approaches, Evaluating communities

UNIT- III: Predictive Analytics in Social Networks

Link prediction problem, Link prediction measures, Feature based Link Prediction, Evaluation

Node classification problem

Node classification: Problem definition and applications; Iterative classification methods; Label propagation method; Graph regularization method; Evaluation

Motif analysis: Definition of network motifs; Triangle counting and enumeration algorithms; Applications of network motifs

UNIT V: Current Research in Social Networks

Social Influence Analysis, privacy in social networks, integrating sensors and social networks, multimedia information networks in social media and social tagging and applications.

Text/Reference books

- Newman, M. E. J. (2010). *Networks: an introduction*. Oxford; New York: Oxford University Press.
- Aggarwal, C. C. (2011). An introduction to social network data analytics. In *Social network data analytics* (pp. 1-15). Springer, Boston, MA.
- Barabási, A. L. (2013). Network science. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *371*(1987), 20120375.

COURSE CODE	COURSE NAME	CORE/ ELECTIVE	CREDITS			
			L	T	P	C
CSE 328	Recommender Systems	E	3	0	0	3

UNIT 1: Introduction
Introduction to Recommender Systems, Applications of Recommender Systems, Goals of Recommender Systems, Basic Models of Recommender Systems, Domain-Specific Challenges in Recommender Systems.

UNIT 2: Neighborhood-Based Recommender Systems
Introduction, Key Properties of Ratings Matrices, Predicting Ratings with Neighborhood-Based Methods: User-Based Neighborhood Models, Item-Based Neighborhood Models, Strengths and Weaknesses of Neighborhood-Based Methods, Dimensionality Reduction and Neighborhood Methods, A Regression Modelling View of Neighborhood Methods, Graph Models for Neighborhood-Based Methods

UNIT 3: Model-Based Collaborative Filtering
Latent Factor Models: Geometric Intuition for Latent Factor Models, Low-Rank Intuition for Latent Factor Models, Basic Matrix Factorization Principles, Unconstrained Matrix Factorization, Singular Value Decomposition, Non-negative Matrix Factorization, Understanding the Matrix Factorization Family

UNIT 4: Content-based Recommender Systems
Basic Components of Content-Based Systems, Feature Extraction: Example of Product Recommendation, Web Page Recommendation, Music Recommendation, Feature Representation and Cleaning, Learning User Profiles and Filtering, Hybrid recommender systems.

UNIT 5: Evaluating Recommender Systems
General goals of evaluation design: accuracy, coverage, confidence and trust, novelty, serendipity, diversity and scalability, Design Issues in Offline Recommender Evaluation, Accuracy Metrics: RMSE, MAE, Evaluating Ranking via Correlation, Evaluating Ranking via Utility, Evaluating Ranking via Receiver Operating Characteristic, Limitations of Evaluation Measures.

References:

1. C.C. Aggarwal, *Recommender Systems: The Textbook*, Springer, 2016.
2. F. Ricci, L Rokach, B. Shapira and P.B. Kantor, *Recommender systems handbook*, Springer 2010.
3. Falk, Kim. Practical recommender systems. Simon and Schuster, 2019.