ANNAMACHARYA UNIVERSITY

(Estd A.P. Private Universities (Establishment and Regulation) Act, 2016)

RAJAMPET-516126, Annamayya District, A.P, INDIA

ACADEMIC REGULATIONS (AU24), COURSE STRUCTURE AND SYLLABI

For the students admitted to

B.Sc., Honors Four-Year / Regular three- year Undergraduate Degree Programme from the Academic Year 2024-25

B.Sc. in Computer Science

B.Sc. – I Year I Semester

Sl. No.	Category	Course Code	Course Title	Hours per Week			Credits
				L	T	P	C
1	BS	24BMAT11T	Matrices and Calculus	3	0	0	3
2	HSM	24BENG11T	Communicative English	2	1	0	3
3	PC	24BCOM11T	Problem Solving through C Programming	3	0	0	3
4	PC	24BCOM12T	Computer Organization	3	0	0	3
5	HSM	24BUHV11T	Human Values	3	0	0	3
6	HSM	24BENG11L	Communicative English Lab	0	0	4	2
7	PC	24BCOM11L	Problem Solving through C Programming Lab	0	0	4	2
Total				**14**	**1**	**8**	**19**

Category	Credits
Professional Core Courses	**8**
Basic Science Course	**3**
Humanities & Social Sciences and Management Courses	**8**

B.Sc.– I Year II Semester

Sl. No.	Category	Course Code	Course Title	Hours per Week			Credits
				L	T	P	C
1	BS	24BMAT21T	Probability Theory and Statistics	3	0	0	3
2	PC	24BCOM21T	Introduction to Python Programming	2	1	0	3
3	PC	24BCOM22T	Object Oriented Programming using C++	3	0	0	3
4	PC	24BCOM23T	Data warehousing & Data Mining	3	0	0	3
5	BS	24BCHE21T	Environmental Studies	3	0	0	3
6	PC	24BCOM21L	Python Programming Lab	0	0	4	2
7	PC	24BCOM22L	Object Oriented Programming using C++ Lab	0	0	4	2
Total				14	**1**	**8**	**19**

Category	Credits
Professional Core Courses	**13**
Basic Science Courses	**6**

ANNAMACHARYA UNIVERSITY

(ESTD UNDER AP PRIVATE UNIVERSITIES (ESTABLISHMENT AND REGULATION) ACT, 2016)

RAJAMPET, Annamayya District, AP – 516126, INDIA

B.Sc. in Computer Science

B.Sc. – II Year I Semester

Sl. No.	Category	Course Code	Course Title	Hours per Week			Credits
				L	T	P	C
1	PC	24BCOM31T	Operating Systems	3	0	0	3
2	PC	24BCOM32T	Software Engineering	3	0	0	3
3	BS	24BMAT31T	Discrete Mathematics	3	0	0	3
4	PC	24BCOM33T	Object Oriented Modeling and Design	3	0	0	3
5	PC	24BCOM34T	Computer Networks	3	0	0	3
6	MC	24BENG31T	Essence of Indian Traditional Knowledge	2	0	0	0
7	PC	24BCOM32L	Software Engineering Lab	0	0	4	2
8	PC	24BCOM33L	UML Lab	0	0	4	2
Total				**17**	**0**	**8**	**19**

Category	Credits
Professional Core Courses	**16**
Basic Science Course	**3**
Mandatory Course	**0**

B.Sc.– II Year II Semester

Sl. No.	Category	Course Code	Course Title	Hours per Week			Credits
				L	T	P	C
1	PC	24BCOM41T	Web Technologies	3	0	0	3
2	PC	24BCOM42T	Software Project Management	3	0	0	3
3	PC	24BCOM43T	Design and Analysis of Algorithms	3	0	0	3
4	BS	24BMAT41T	Numerical Methods	3	0	0	3
5	PC	24BCOM44T	Internet of Things	3	0	0	3
6	MC	24BENG41T	Constitution of India.	2	0	0	0
7	PC	24BCOM41L	Web Technologies Lab	0	0	4	2
8	PC	24BCOM43L	Design and Analysis of Algorithms Lab	0	0	4	2
Total				17	**0**	**8**	**19**

Category	Credits
Professional Core Courses	**16**
Basic Science Course	**3**
Mandatory Course	**0**

B.Sc. in Computer Science

B.Sc. – III Year I Semester

Sl. No.	Category	Course Code	Course Title	Hours per Week			Credits
				L	T	P	C
1	PC	24BAIL54T	Introduction to Artificial Intelligence	3	0	0	3
2	PC	24BAIA53T	Fundamentals of Big Data	3	0	0	3
3	PC	24BCOM51T	Advanced Python Programming	3	0	0	3
4	BS	24BMAT51T	Differential Equations	3	0	0	3
5	PE	24BCOM5AT	Software Architecture	3	0	0	3
		24BCOM5BT	Quantum Computing				
		24BCOM5CT	Natural Language Processing				
6	PC	24BAIL54L	Artificial Intelligence Lab	0	0	4	2
7	PC	24BCOM51L	Advanced Python Programming Lab	0	0	4	2
8	SEC	24BENG51S	Soft Skills	1	0	2	2
9	OE	24BCOM5DT	MOOC	3	0	0	3
Total				**15**	**0**	**08**	**24**

Category	Credits
Professional Core Courses	**13**
Professional Elective Courses	**3**
Basic Science Course	**3**
Skill Enhancement Course	**2**
Open Elective (MOOC)	**3**

B.Sc.– III Year II Semester

Sl. No.	Category	Course Code	Course Title	Hours per Week			Credits
				L	T	P	C
1	PE	24BCOM6AT	Distributed Systems	3	0	0	3
		24BCOM6BT	Web Services				
		24BCOM6CT	Information Retrieval Systems				
2	BS	24BMAT61T	Transform Techniques	3	0	0	3
3	INTERN	24BCOM61I	Summer Internship 6 weeks after second year (to be evaluated VI Semester)	0	0	0	2
4	PROJ	24BCOM62P	Project Work	0	0	0	12
Total				06	**0**	**15**	**20**

Category	Credits
Professional Elective Courses	**3**
Basic Science Course	**3**
Internship	**2**
Project Work	**12**

Title of the Course: Matrices and Calculus
Category: BS
Course Code: 24BMAT11T
Branches: **Computer Science, AI&ML, AI&DA, Programming & Data Science**
Semester: I Semester

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	0	0	3

Course Objectives: The goal is to familiarize oneself with matrix theory and its applications, gain a solid understanding of partial derivatives and mean value theorems to tackle mathematical problems, and develop proficiency in vector differentiation and integration to address complex challenges effectively.

Course Outcomes:
At the end of the course, the student will be able to
1. understand the methods for solving the system of linear equations
2. utilize matrix algebra techniques for engineering applications.
3. analyze the functions of several variables to optimization techniques.
4. use scalar and vector point functions across a range of technical disciplines.
5. apply vector integral theorems in evaluating double and triple integrals

Unit 1 Matrices 12

Rank of a matrix by echelon form, normal form, solving system of homogeneous and non-homogeneous linear equations, Cayley-Hamilton theorem (without proof), finding inverse and power of a matrix by Cayley-Hamilton theorem.

Unit 2 Eigen values and Eigen vectors 8

Eigen values and Eigen vectors and their properties, diagonalization of a matrix, Quadratic forms and nature of the quadratic forms, reduction of quadratic form to canonical form by orthogonal transformation.

Unit 3 Mean Value Theorems & Multivariable calculus 8

Taylor's theorem and Maclaurin's theorem for one variable (without proofs) – Simple problems.
Partial derivatives, total derivatives, chain rule, change of variables, Jacobian, Maxima and Minima of functions of two variables, method of Lagrange multipliers for three variables

Unit 4 Vector Differential Calculus 10

Scalar and vector point functions, vector operator Del, Gradient, Directional derivative, Divergence, Curl, Del applied twice to point functions, vector identities. Applications: Irrotational fields and solenoidal fields.

Unit 5 Vector Integral Calculus 10

Line integral, surface integral and volume integral. Green's theorem in the plane (without proof), verifications of Stroke's theorem (without proof) and Gauss's divergence theorem (without proof).

Prescribed Textbooks:

1. B.S. Grewal. *Higher Engineering Mathematics.* 44th Ed., Khanna Publishers, 2017.
2. J. Stewart. *Calculus: Early Transcendentals.* 7th Ed., Cengage Learning.

Reference Books:

1. G. B. Thomas, Maurice D. Weir and Joel Hass. *Thomas' Calculus*, 14th Ed., Pearson Publishers, 2022.
2. R.K. Jain, S.R.K. Iyengar. *Advanced Engineering Mathematics.* 5th Ed., Narosa Publications.

Title of the Course: Communicative English
Category: HSM
Course Code: 24BENG11T
Branches: **Computer Science, AI&ML, AI&DA, Programming & Data Science**
Semester: I Semester

Lecture Hours	Tutorial Hours	Practice Hours	Credits
2	1	0	3

Course Objectives: The primary goal of the Communicative English course is to develop students' listening, reading, speaking, and writing skills, enhancing their comprehension, oral presentation, and reporting abilities. It focuses on improving grammatical structures and vocabulary, equipping students with the proficiency needed for effective communication and preparing them to meet industry standards.

Course Outcomes:
At the end of the course, the student will be able to
1. understand the context, topic, and pieces of specific information from social or transactional dialogues spoken by native speakers of English
2. analyze literary forms, journalistic articles and scientific readings for comprehension and retention
3. demonstrate effective writing and speaking skills
4. apply grammatical knowledge in speech and writing and formulate sentences with accuracy
5. produce coherent and unified paragraphs with adequate support and detail

Unit 1 HUMAN VALUES: The Gift of the Magi (Short Story) 12

Listening: Identifying the topic, the context and specific pieces of information by listening to short audio texts and answering a series of questions.
Speaking: Asking and answering general questions on familiar topics such as home, family, work, studies and interests; introducing oneself and others.
Reading: Skimming to get the main idea of a text; scanning to look for specific pieces of information.
Writing: Mechanics of Writing-Capitalization, Spellings, Punctuation-Parts of Sentences.
Grammar: Parts of Speech, Basic Sentence Structures-forming questions
Vocabulary: Synonyms, Antonyms, Affixes (Prefixes/Suffixes), Root words.

Unit 2 NATURE: The Brook by Alfred Tennyson (Poem) 8

Listening: Answering a series of questions about main ideas and supporting ideas after listening to audio texts.
Speaking: Discussion in pairs/small groups on specific topics followed by short structure talks.
Reading: Identifying sequence of ideas; recognizing verbal techniques that help to link the ideas in a paragraph together.
Writing: Structure of a paragraph - Paragraph writing (specific topics)
Grammar: Cohesive devices - linkers, use of articles and zero article; prepositions.
Vocabulary: Homonyms, Homophones, Homographs.

Unit 3 BIOGRAPHY: Elon Musk 8

Listening: Listening for global comprehension and summarizing what is listened to.
Speaking: Discussing specific topics in pairs or small groups and reporting what is discussed
Reading: Reading a text in detail by making basic inferences -recognizing and interpreting specific context clues; strategies to use text clues for comprehension.
Writing: Summarizing, Note-making, paraphrasing
Grammar: Verbs - tenses; subject-verb agreement; Compound words, Collocations
Vocabulary: Compound words, Collocations

Unit 4 INSPIRATION: The Toys of Peace by Saki 10

Listening: Making predictions while listening to conversations/ transactional dialogues without video; listening with video.
Speaking: Role plays for practice of conversational English in academic contexts (formal and informal) - asking for and giving information/directions.
Reading: Studying the use of graphic elements in texts to convey information, reveal trends/patterns/relationships, communicate processes or display complicated data.
Writing: Letter Writing: Official Letters, Resumes
Grammar: Reporting verbs, Direct & Indirect speech, Active & Passive Voice
Vocabulary: Words often confused, Jargons

Unit 5 MOTIVATION: The Power of Intrapersonal Communication (An Essay) 10

Listening: Identifying key terms, understanding concepts and answering a series of relevant questions that test comprehension.
Speaking: Formal oral presentations on topics from academic contexts
Reading: Reading comprehension.
Writing: Writing structured essays on specific topics.
Grammar: Editing short texts –identifying and correcting common errors in grammar and usage (articles, prepositions, tenses, subject verb agreement)
Vocabulary: Technical Jargons

Prescribed Textbooks:

1. *Pathfinder: Communicative English for Undergraduate Students.* 1st Ed., Orient Black Swan, 2023 (Units 1,2 & 3)

Reference Books:

1. Dubey, Sham Ji & Co. *English for Engineers.* Vikas Publishers, 2020
2. Bailey, Stephen. *Academic writing: A Handbook for International Students*. Routledge, 2014.
3. Murphy, Raymond. *English Grammar in Use.* 4th Ed., Cambridge University Press, 2019.
4. Lewis, Norman. *Word Power Made Easy- The Complete Handbook for Building Superior Vocabulary.* Anchor, 2014.

Web Resources:
GRAMMAR:

1. www.bbc.co.uk/learningenglish
2. https://dictionary.cambridge.org/grammar/british-grammar/
3. www.eslpod.com/index.html
4. https://www.learngrammar.net/
5. https://english4today.com/english-grammar-online-with-quizzes/
6. https://www.talkenglish.com/grammar/grammar.aspx

VOCABULARY

1. https://www.youtube.com/c/DailyVideoVocabulary/videos
2. https://www.youtube.com/channel/UC4cmBAit8i_NJZE8qK8sfpA

ANNAMACHARYA UNIVERSITY

(ESTD UNDER AP PRIVATE UNIVERSITIES (ESTABLISHMENT AND REGULATION) ACT, 2016)
(UNIVERSITY LISTED IN UGC AS PER THE SECTION 2(f) OF THE UGC ACT, 1956)

RAJAMPET, Annamayya District, AP – 516126, INDIA

Title of the Course: Problem Solving through C Programming
Category: PC
Course Code: 24BCOM11T
Branches: **Computer Science, AI&ML, AI&DA, Programming & Data Science**
Semester: I Semester

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	0	0	3

Course Objectives:

1. Understanding the steps in problem solving and formulation of algorithms to problems.
2. Develop programming skills as a means of implementing an algorithmic solution with appropriate control and data structures.
3. Develop intuition to enable students to come up with creative approaches to problems.
4. Develop programs using pointers, structures and unions
5. Manipulation of text data using files

Course Outcomes:

At the end of the course, the student will be able to

1. formulate solutions to problems and represent those using
2. choose proper control statements and use arrays for solving problems.
3. decompose a problem into modules and use functions to implement the
4. apply and use allocation of memory for pointers and solve the problems
5. develop the solutions for problems using C programming Language.

Unit 1 Problem Solving and Introduction to C 12

Steps to solve problems, algorithm, Pseudo code, Flowchart with examples, Program Development Environments. Introduction to programming: Programming languages and generations.
Introduction to C: Introduction, structure of C program, keywords, identifiers, Variables, data types, constants, I/O statements, operators, precedence and associatively.

Unit 2 Introduction to decision control statements and Arrays 8

Selective, looping and nested statements, jumping statements.
Arrays: Introduction, declaration of arrays, accessing and storage of array elements, searching (linear and binary search algorithms) and sorting (selection and bubble) algorithms, multidimensional arrays, matrix operations

Unit 3 Strings and Functions 8

Strings: Declaration and Initialization, String Input / Output functions, String manipulation functions.
Functions: Types of functions, recursion, scope of variables and storage classes.
Preprocessor Directives: Types of preprocessor directives, examples.

Unit 4 Pointers 10

Pointers: Understanding computer's memory, introduction to pointers, declaration pointer variables, pointer arithmetic, pointers and strings, array of pointers, function pointers, dynamic memory allocation, advantages and drawbacks of pointers

Unit 5 Structures and Files 10

Structures: Structure definition, initialization and accessing the members of a structure, nested structures, array of structures, structures and functions, structures and pointers, self-referential structures, unions and enumerated data types.

Files: Introduction to files, file operations, reading and writing data on files, error handling during file operations.

Prescribed Textbooks:

1. B.A. Forouzan, R. F. Gilberg. *C Programming and Data Structures*. Cengage learning, Indian edition.
2. E. Balagurusamy. *C and Data Structures*. Tata McGraw Hill.
3. J. R. Hanly, Ashok N. Kamthane and A. Ananda Rao. *Programming in C and Data Structures*. Pearson Education.

Reference Books:

1. Yashvanth Kanetkar. *LET US C.* 9th Ed., BPB Publication
2. Byron Gottfried. *Schaum's Outline of Programming with C*. McGraw-Hill.
3. Brian W. Kernighan and Dennis M. Ritchie. *The C Programming Language*. Prentice Hall of India.
4. A. K. Sharma. *Computer Fundamentals and Programming.* 2nd Ed., University Press, 2018.
5. Pradeep Dey and Manas Ghosh. *Programming in C*. 2nd Ed., Oxford Press, 2017

Title of the Course: Computer Organization
Category: PC
Course Code: 24BCOM12T
Branch/es: **Computer Science, AI&ML, AI&DA, Programming & Data Science**
Semester: I Semester

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	0	0	3

Course Objectives:

1. Illuminate the student to understand the basic concepts of digital number systems and their conversions
2. Allow the students to Design and analyze combinational and sequential logic circuits through formulation of logic functions, Boolean algebra minimization
3. Impart the students to understand the internal organization and operations of a computer
4. Enable the students to acquire knowledge about the concepts of processor logic design and memory organization
5. Allow the students to familiarize the concepts related to IO organization.

Course Outcomes:

At the end of the course, the student will be able to

1. identify the basic structure and functional units of a digital computer
2. solve problems based on computer arithmetic
3. design, Analyze and evaluate different digital circuits using Boolean algebra
4. understand instruction structure and analyze the effect of addressing modes on the execution time of a program
5. understand concepts related to Processor, memory organization and select appropriate interfacing standards for I/O devices

Unit 1 Basic Structure of Computers 12

Computer Types, Functional units, Basic operational concepts, Bus structures, Data Representation: Binary Numbers, Fixed Point Representation. Floating – Point Representation. Number base conversions, Octal and Hexadecimal Numbers, complements, Signed binary numbers, Binary codes.

Unit 2 Digital Logic Circuits 8

Digital Logic Circuits - I: Basic Logic Functions, Logic gates, universal logic gates, Minimization of Logic expressions, Combinational Circuits.
Digital Logic Circuits - II: Flip-Flops, Registers, Shift Registers, Binary counters, Decoders, Multiplexers, Programmable Logic Devices.

Unit 3 Computer Arithmetic and Instruction Set & Addressing 8

Computer Arithmetic: Algorithms for fixed point and floating-point addition, subtraction, multiplication and division operations, Hardware Implementation of arithmetic and logic operations.

Instruction Set & Addressing: Memory Locations and Addresses, Machine addresses and sequencing, Various Addressing Modes, Instruction Formats, Basic Machine Instructions.

Unit 4 Processor Organization and Memory Organization 10

Processor Organization: Introduction to CPU, Register Transfers, Execution of Instructions, Multiple Bus Organization, Hardwired Control, Micro programmed Control

Memory Organization: Concept of Memory, RAM, ROM memories, memory hierarchy, cache memories, virtual memory, secondary storage, memory management hardware.

Unit 5 Input / Output Organization 10

Introduction to I/O, Interrupts- Hardware, Enabling and disabling Interrupts, Device Control, Direct memory access, buses, Interface Circuits, Standard I/O Interfaces.

Prescribed Textbooks:

1. Carl Hamacher, Zvonko Vranesic, Safwat Zaky. *Computer Organization.* 5th Ed., McGraw Hill.
2. Miles Murdocca, Vincent Heuring. *Computer Architecture and Organization- An Integrated Approach.* 2nd Ed., Wiley India.
3. M. Moris Mano. *Computer System Architecture.* 3rd Ed., Pearson.

Reference Books:

1. William Stallings. *Computer Organization and Architecture*. 6th Ed., Pearson.
2. David A. Paterson and John L. Hennessy. *Computer- organization and Design*. Elsevier.
3. Sivarama Dandamudi. *Fundamentals or Computer Organization and Design*. Springer Int. Edition.
4. M. Morris Mano. *Digital* Design. 3rd Ed., Pearson Education/PHI.

Title of the Course: Human Values
Category: HSM
Course Code: 24ENG11T
Branches: **Computer Science, AI&ML, AI&DA, Programming & Data Science**
Semester: I Semester

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	0	0	3

Course Objectives: The course aims to help students recognize the vital interplay between values and skills, promoting sustained happiness and prosperity. It seeks to foster a holistic perspective on life and profession, encouraging value-based living through a clear understanding of human reality and existence. Additionally, the course emphasizes the ethical implications of such an understanding, fostering trustful behaviour, fulfilling human interactions, and a harmonious relationship with nature.

Course Outcomes:
At the end of the course, the student will be able to
1. comprehend the terms like Natural Acceptance, Happiness and Prosperity
2. analyze one's self, and one's surroundings (family, society, nature)
3. apply human values in enriching human relationships and human society.
4. analyze the need for universal human values and harmonious existence.
5. evaluate themselves as socially and ecologically responsible engineers.

Unit 1 Introduction to Value Education 12

Lecture1: Right Understanding, Relationship and Physical Facility (Holistic Development and the Role of Education)
Lecture2: Understanding Value Education
Tutorial 1: Practice Session1: Sharing about oneself
Lecture 3: Self-Exploration as the Process for Value Education
Lecture4: Continuous Happiness and Prosperity – The Basic Human Aspirations
Tutorial 2: Practice Session2: Exploring Human Consciousness
Lecture 5: Happiness and Prosperity – Current Scenario
Lecture 6: Method to Fulfill the Basic Human Aspirations
Tutorial 3: Practice Session3: Exploring Natural Acceptance

Unit 2 Harmony in the Human Being 8

Lecture 7: Understanding Human being as the co-existence of the self and the body.
Lecture 8: Distinguishing between the needs of the self and the body
Tutorial 4: Practice Session4: Exploring the difference of needs of self and body.
Lecture 9: The body as an instrument of the self
Lecture 10: Understanding harmony in the self
Tutorial 5: Practice Session5: Exploring sources of imagination in the self
Lecture 11: Harmony of the self with the body

Lecture 12: Programme to ensure self-regulation and health
Tutorial 6: Practice Session 6: Exploring harmony of self with the body

Unit 3 Harmony in the Family and Society 8

Lecture 13: Harmony in the Family – the Basic Unit of Human Interaction
Lecture 14: 'Trust' – the Foundational Value in Relationship
Tutorial 7: Practice Session7: Exploring the Feeling of Trust
Lecture 15: 'Respect' – as the Right Evaluation
Tutorial 8: Practice Session 8: Exploring the Feeling of Respect
Lecture 16: Other Feelings, Justice in Human-to-Human Relationship
Lecture 17: Understanding Harmony in the Society
Lecture 18: Vision for the Universal Human Order
Tutorial 9: Practice Session9: Exploring systems to fulfil human goal

Unit 4 Harmony in the Nature/Existence 10

Lecture 19: Understanding Harmony in the Nature
Lecture 20: Interconnectedness, Self-Regulation and Mutual Fulfillment among the Four Orders of Nature
Tutorial 10: Practice Session 10: Exploring the Four Orders of Nature
Lecture 21: Realizing Existence as Co-existence at All Levels
Lecture 22: The Holistic Perception of Harmony in Existence
Tutorial 11: Practice Session11: Exploring Co-existence in Existence

Unit 5 Implications of the Holistic Understanding – a Look at Professional Ethics 10

Lecture 23: Natural Acceptance of Human Values
Lecture 24: Definitiveness of (Ethical) Human Conduct
Tutorial 12: Practice Session 12: Exploring Ethical Human Conduct
Lecture 25: A Basis for Humanistic Education, Humanistic Constitution and Universal Human Order
Lecture 26: Competence in Professional Ethics
Tutorial 13: Practice Session13: Exploring Humanistic Models in Education
Lecture 27: Holistic Technologies, Production Systems and Management
Models-Typical Case Studies
Lecture 28: Strategies for Transition towards Value-based Life and Profession
Tutorial 14: Practice Session14: Exploring Steps of Transition towards Universal Human Order

Practice Sessions for UNIT I – Introduction to Value Education
PS1 Sharing about Oneself
PS2 Exploring Human Consciousness
PS3 Exploring Natural Acceptance
Practice Sessions for UNIT II – Harmony in the Human Being
PS4 Exploring the Difference of Needs of Self and Body
PS5 Exploring Sources of Imagination in the Self
PS6 Exploring Harmony of Self with the Body
Practice Sessions for UNIT III – Harmony in the Family and Society
PS7 Exploring the Feeling of Trust
PS8 Exploring the Feeling of Respect
PS9 Exploring Systems to fulfil Human Goal
Practice Sessions for UNIT IV – Harmony in the Nature (Existence)
PS10 Exploring the Four Orders of Nature
PS11 Exploring Co-existence in Existence
Practice Sessions for UNIT V – Implications of the Holistic Understanding – a Look at Professional Ethics
PS12 Exploring Ethical Human Conduct

PS13 Exploring Humanistic Models in Education
PS14 Exploring Steps of Transition towards Universal Human Order

Prescribed Textbooks:

1. **The Textbook** - R R Gaur, R Asthana, G P Bagaria. *A Foundation Course in Human Values and Professional Ethics.* 2nd Revised Edition, Excel Books, New Delhi, 2019. ISBN 978-93-87034-47-1.
2. The Teacher's Manual - R R Gaur, R Asthana, G P Bagaria. *Teachers' Manual for A Foundation Course in Human Values and Professional Ethics.* 2nd Revised Edition, Excel Books, New Delhi, 2019. ISBN 978-93-87034-53-2

Reference Books:

1. A. Nagaraj. *Jeevan Vidya: Ek Parichaya.* Jeevan Vidya Prakashan, Amarkantak, 1999.
2. A.N. Tripathi. *Human Values.* New Age Intl. Publishers, New Delhi, 2004.
3. The Story of Stuff (Book).
4. Mohandas Karamchand Gandhi. *The Story of My Experiments with Truth*
5. E. F Schumacher. *Small is Beautiful*

Web Resources:

1. https://fdp-si.aicte-india.org/UHV-II%20Class%20Notes%20&%20Handouts/UHV%20Handout%201Introduction%20to%20Value%20Education.pdf
2. https://fdp-si.aicte-india.org/UHV-II%20Class%20Notes%20&%20Handouts/UHV%20Handout%202-Harmony%20in%20the%20Human%20Being.pdf
3. https://fdp-si.aicte-india.org/UHV%201%20Teaching%20Material/D3S2%20Respect%20July%2023.pdf
4. https://fdp-si.aicte-india.org/UHV-II%20Class%20Notes%20&%20Handouts/UHV%20Handout%205-Harmony%20in%20the%20Nature%20and%20Existence.pdf
5. https://fdp-si.aicte-india.org/UHV%20II%20Teaching%20Material/UHV%20II%20Lecture%202325%20Ethics%20v1.pdf
6. https://www.studocu.com/in/document/kiet-group-of-institutions/universal-humanvalues/chapter-5-holistic-understanding-of-harmony-on-professional-ethics/62490385https://onlinecourses.swayam2.ac.in/aic22_ge23/preview
7. https://fdp-si.aicte-india.org/UHV-II%20Class%20Notes%20&%20Handouts/UHV%20Handout%203-Harmony%20in%20the%20Family.pdf

Title of the Course: Communicative English Lab
Category: HSM
Course Code: 24BENG11L
Branches: **Computer Science, AI&ML, AI&DA, Programming & Data Science**
Semester: I Semester

Lecture Hours	Tutorial Hours	Practice Hours	Credits
0	0	4	2

Course Objectives:
The main objective of Communicative English Laboratory course is to provide students with a range of self-instructional and learner-friendly methods for enhancing their language skills. This course is designed to train students in fundamental communication abilities and prepare them effectively for job interviews.

Course Outcomes:
At the end of the course, the student will be able to
1. relate and differentiate the English speech sounds, stress, rhythm, intonation and syllable division for better listening and speaking comprehension
2. develop effective communication skills through various verbal and non-verbal language learning practices
3. demonstrate professional skills by actively participating role-plays and descriptions
4. build an effective resonate and equip themselves with essential employability skills for professional success
5. enhance public speaking skills and deliver oral presentations with clarity and confidence

List of Topics
1. Phonetics: Vowels & Consonants
2. Just A Minute (JAM)
3. Role Play / Situational Dialogues
4. Oral Presentation/ Poster Presentation
5. Information Transfer
6. Describing people/objects/situations

Suggested Software:
1. Walden Infotech
2. Sky Pronunciation Suite
3. Clarity Pronunciation Power – Part I
4. Learning to Speak English - 4 CDs
5. Loose Your Accent in 28 days, CD Rom, Judy Ravin

Prescribed Textbooks:

1. Meenakshi Raman, Sangeeta-Sharma. *Technical Communication*. Oxford Press.2018.
2. Grant Taylor. *English Conversation Practice.* Tata McGraw-Hill Education India, 2016

3. Hewing's, Martin. *Cambridge Academic English* (B2). CUP, 2012.
4. T. Balasubramanyam. *A Textbook of English Phonetics for Indian Students.* 3rd Ed., Trinity Press.

Web Resources:

Spoken English:

1. www.esl-lab.com
2. www.englishmedialab.com
3. www.englishinteractive.net
4. https://www.britishcouncil.in/english/online
5. http://www.letstalkpodcast.com/
6. https://www.youtube.com/c/mmmEnglish_Emma/featured
7. https://www.youtube.com/c/ArnelsEverydayEnglish/featured
8. https://www.youtube.com/c/engvidAdam/featured
9. https://www.youtube.com/c/EnglishClass101/featured
10. https://www.youtube.com/c/SpeakEnglishWithTiffani/playlists
11. https://www.youtube.com/channel/UCV1h_cBE0Drdx19qkTM0WNw

Voice & Accent:

1. https://www.youtube.com/user/letstalkaccent/videos
2. https://www.youtube.com/c/EngLanguageClub/featured
3. https://www.youtube.com/channel/UC_OskgZBoS4dAnVUgJVexc
4. https://www.youtube.com/channel/UCNfm92h83W2i2ijc5Xwp_IA

Title of the Course: Problem Solving through C Programming Lab
Category: PC
Course Code: 24ACOM11L
Branches: **Computer Science, AI&ML, AI&DA, Programming & Data Science**
Semester: I Semester

Lecture Hours	Tutorial Hours	Practice Hours	Credits
0	0	4	2

Course Objectives:

1. Setting up programming environment.
2. Develop Programming skills to solve problems.
3. Use of appropriate C programming constructs to implement algorithms.
4. Identification and rectification of coding errors in program
5. Develop applications using modular programming and Manage data using files.

Course Outcomes:

At the end of the course, the student will be able to

1. Identify and setup program development environment
2. Implement the algorithms using C programming language constructs
3. Identify and rectify the syntax errors and debug program for semantic errors
4. Solve problems in a modular approach using functions
5. Implement file operations with simple text data

Unit 1 Data Types, Constants, Input and Output and expressions 12

Exercise 1: Data types, Variables, Constants and Input and Output.
Exercise 2: Operators, Expressions and Type Conversions.

Unit 2 Decision Control Statements and Arrays 8

Exercise 3: Conditional Statements [two way and multipath].
Exercise 4: Loop Control Statements. [for, while and do-While]
Exercise 5: Unconditioned JUMP Statements- break, continue, goto.
Exercise 6: Declaring Arrays, Referencing Arrays, Array Subscripts. Using for loop for sequential Access.
Exercise 7: Multidimensional Arrays

Unit 3 Strings and Functions 8

Exercise 8: String Basics, String Library Functions and Array of Strings.
Exercise 9: Simple user defined functions, Parameter passing methods- pass by value, pass by reference.
Exercise 10: Storage classes- Auto, Register, Static and Extern
Exercise 11: Recursive Functions, Preprocessor commands.
Exercise 12: Array Elements as Function Arguments.

Unit 4 Pointers 10

Exercise 13: Pointers, Dynamic memory allocation and error handling

Unit 5 Structures and Files 10

Exercise 14: Structures

Exercise 15: File handling

Prescribed Textbooks:

1. B.A. Forouzan, R. F. Gilberg. *C Programming and Data Structures.* Cengage learning, Indian edition.
2. E. Balagurusamy. *C and Data Structures.* Tata McGraw Hill.
3. J. R. Hanly, Ashok N. Kamthane and A. Ananda Rao. *Programming in C and Data Structures.* Pearson Education.

Reference Books:

1. Yashvanth Kanetkar. *Let Us C*, 9th Ed., BPB Publication
2. A K Sharma. *Computer Fundamentals and Programming.* 2nd Ed., University Press, 2018.
3. Pradeep Dey and Manas Ghosh. *Programming in C.* Oxford Press, 2017
4. Reema Thareja, *Introduction to C Programming*, 2nd Ed., OXFORD Press, 2015
5. https://www.cprogramming.com/
6. https://www.mycplus.com/tutorials/c-programming-tutorials

Title of the Course: Probability Theory and Statistics
Category: BS
Course Code: 24BMAT21T
Branches: **Computer Science, AI&ML, AI&DA, Programming & Data Science**
Semester: II Semester

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	0	0	3

Course Objectives: The aim is to introduce the fundamental concepts of probability theory and elucidate probability distributions for problem-solving. This includes describing measures of central tendency, explaining hypothesis testing and confidence intervals for large samples, and detailing the steps for hypothesis testing with small samples.

Course Outcomes:
At the end of the course, the student will be able to
1. describe the fundamental concepts of probability and random variables.
2. apply the probability distribution in real life problems
3. calculate the correlation between two variables
4. Apply estimation techniques and conduct hypothesis tests for large samples.
5. Analyze the effectiveness of various hypothesis testing methods for small samples to derive accurate conclusions

Unit 1 Probability and Random Variables 10

Probability - axioms of probability – addition theorem of probability - conditional probability- multiplication theorem of probability (without proof) - Baye's theorem.
Random variables - discrete and continuous - Distribution functions - Mean and Variance.

Unit 2 Probability distributions 6

Probability distributions – Binomial and Poisson distribution - fitting - normal distribution - their properties.

8

Unit 3 Introduction of statistics

Introduction of Statistics – Mean - Median and Mode for ungrouped and grouped data.
Correlation - correlation coefficient – Karl Pearson's coefficient - Spearman's rank correlation.

Unit 4 Estimation and Testing of hypothesis for large samples 10

Estimation - Point estimation - Interval estimation of one mean (small and large) - one Proportion (large).
Test of Hypothesis: Types of errors, one and two tailed tests, level of significance, single mean -difference of means - single proportion - difference of proportions (large).

Unit 5 Testing of hypothesis for small samples 10

Student t-distribution test for single mean - two means and paired t-test,
Testing of equality of variances (F-test) - χ^2 test for goodness of fit - χ^2 test for independence of attributes.

Prescribed Textbooks:

1. Miller and Freund's. *Probability and Statistics for Engineers.*7th Ed., Pearson, 2008.
2. S.C. Gupta and V.K. Kapoor. *Fundamentals of Mathematical Statistics.* 11th Ed., Sultan Chand & Sons Publications, 2012.

Reference Books:

1. S. Ross. *a First Course in Probability.* Pearson Education India, 2002.
2. W. Feller. *An Introduction to Probability Theory and its Applications.* 1st Ed., Wiley, 1968.

Title of the Course: Introduction to Python Programming
Category: PC
Course Code: 24BCOM21T
Branches: **Computer Science, AI&ML, AI&DA, Programming & Data Science**
Semester: II Semester

Lecture Hours	Tutorial Hours	Practice Hours	Credits
2	1	0	3

Course Objectives:

1. To learn basics of computational problem solving, python programming and basic control structures.
2. To know python programming basic constructs like lists, dictionaries, sets and functions
3. To understand basics of object-oriented programming
4. To understand the performance of the implementations of basic data structures.

Course Outcomes:

At the end of the course, the student will be able to

1. Understand Python basics: literals, variables, operators, and control structures.
2. Develop functions, modules, and process strings; learn object-oriented principles.
3. Apply OOP concepts: inheritance, polymorphism, and handle exceptions effectively.
4. Implement advanced data structures: linked lists, stacks, and queues.
5. Utilize tree structures and algorithms: binary trees, search trees.

Unit 1 12

Introduction to python programming language, literals, variables and identifiers, operators, expressions and data types. Control Structures: Control structure importance, Boolean expressions, selection control, and iterative control.

Data Structures in python: List structures, lists in python, iterating over lists (sequences) in python, more on python lists, Dictionary, Set

Unit 2 8

Functions: Program routines, more functions. Module Design: Modules, Top-Down design, python modules. String Processing: String Traversal, String-Applicable Sequence Operations. String Methods

Introduction to Object oriented programming: class, three fundamental features of object-oriented programming. What is encapsulation? Defining classes in python.

Data abstraction and through classes, special methods, calling a class method from another class method, garbage collection, class and static methods.

Unit 3 8

Inheritance: Introduction, Inheriting classes in python, types of inheritance, abstract classes and interfaces. Polymorphism: Operator overloading: Introduction, implementing operator overloading, method overriding.

Error and Exception handling: introduction, handling exceptions, multiple except blocks, multiple

exceptions in a single block, the else clause, raising exceptions, instantiating exceptions, handling exceptions in invoked functions, built-in and user defined exceptions, the finally block, Assertions in python.

Unit 4 10

Data structures: Introduction to abstract data types, Single Linked List-traversing, searching, prepending, and removing nodes. Stacks-implementing using python list & linked list, Queues-implementing using python list & linked list.

Unit 5 10

Binary Trees: The Tree structure, the binary tree, priority queues-heaps Search trees: The binary search tree, search tree iterators, AVL trees

Prescribed Textbooks:

1. Charles Dierbach. *Introduction to Computer Science Using Python: A Computational Problem-Solving Focus.* Wiley.
2. Rance D. Necaise. *Data Structures and Algorithms using Python.* Wiley Publications.

Reference Books:

1. R. Nageswara Rao. *Core Python Programming.* Dream Tech Press (Wiley India), 2017.
2. Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser. *Data Structures and Algorithms in Python*, Wiley Publications
3. Reema Thareja. *Python Programming using problem solving approach.* Oxford University press
4. John Zelle, Franklin, Beedle. *Python Programming: An Introduction to Computer Science.* 3rd Ed.,
5. Allen Downey. *Think Python: How to think like a computer Scientist.* 2nd Ed., O'Reilly Publications
6. Bradley Miller, David L. Ranum, Franklin, Beedle. *Problem solving with algorithms and data structures using python.* independent publishers

Title of the Course: Object Oriented Programming using C++
Category: PC
Course Code: 24BCOM22T
Branches: **Computer Science, AI&ML, AI&DA, Programming & Data Science**
Semester: II Semester

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	0	0	3

Course Objectives:

1. To learn New & Advanced Data Structures
2. To impart the Object-Oriented Concepts in C++
3. To acquire knowledge on Algorithmic Design and Analysis
4. To solve problems Using Different Data Structures and Design Techniques, and Compare their Performance and Tradeoffs
5. To implement pattern matching algorithms and Tree Data Structures in C++

Course Outcomes:

At the end of the course, the student will be able to

1. Able to understand the basic concepts of C++ and its functions
2. Able to understand and apply the Object-Oriented Concepts and performance analysis for algorithms
3. Able to apply and analyze the abstract data types such as Stacks, Queues, and Dictionaries
4. Able to categorize tree data structures such as binary search trees, AVL, Red-black, Splay trees
5. Able to determine pattern matching algorithm efficiencies.

Unit 1 INTRODUCTION TO C++: 12

Introduction, Class Overview: Class, Objects, Class Members, I/O Streams, Access Control, Class Scope, Static Class Members: Static Member Variables, Static Member Functions, Static Object, Functions: Parameter Passing Methods, Inline Functions, The Friend Function, This Pointer, Dynamic Memory Allocation and Deallocation: New Operator, Delete Operator, Exception Handling.

Unit 2 OBJECT ORIENTED CONCEPTS 8

Constructors, Constructor Overloading, Destructors, Function Overloading, Operator Overloading: Plus, Minus, Unary, Inheritance: Base Class Access Control, Types of Inheritance, Reasons for the usage of Inheritance, Polymorphism: Virtual Functions, Pure Virtual Functions, Abstract Classes, Generic Programming with Templates: Function Templates, Class Templates.
Algorithms: Performance Analysis, Space Complexity, Time Complexity: Bubble Sort, Selection Sort.

Unit 3 STACKS, QUEUES AND DICTIONARIES: 8

Stack ADT, Queue ADT, Operations of Stack & Queue ADT
Dictionaries, Linear List Representation, Skip List Representation: Operations, Searching, Insertion, Deletion, Hash Table: Hash Functions, Collisions: Separate Chaining, Open Addressing - Linear Probing, Quadratic Probing, Double Hashing or Rehashing, Extendible Hashing, Comparison of Chaining and Open

Addressing.

Unit 4 PRIORITY QUEUES, SEARCH TREES (PART I): 10

Priority Queue ADT, Priority Queue Implementation Using Heaps, External Sorting.

Binary Search Trees ADT, Representation of Binary Search Tree, Operations on Binary Search Trees: Insertion, Deletion and Searching, AVL Trees, Operations of AVL Trees: Insertion, Deletion and Searching.

Unit 5 SEARCH TREES (PART II) AND PATTERN MATCHING 10

Introduction to Red–Black and Splay Trees, B-Trees, Operations on B-Trees: Insertion, Deletion and Searching, Height of B Tree.

Pattern Matching Algorithms, Fixed Pattern Matching Algorithms: Brute Force, Boyer–Moore, Knuth-Morris-Pratt Algorithms.

Prescribed Textbooks:

1. Akepogu Ananda Rao, Palagiri Radhika Raju. *Data Structures and Algorithms Using C++.* Pearson Education.
2. Sartaj Sahni. *Data Structures, Algorithms and Applications in C++.* 2nd Ed., Universities Press (India) Pvt. Ltd.
3. Mark Allen Weiss. *Data Structures and Algorithm Analysis in C++.* 2nd Ed, Pearson Education.

Reference Books:

1. Michael T. Goodrich, R. Tamassia and Mount. *Data Structures and Algorithms in C++.* Wiley Student Edition, John Wiley and Sons.
2. Adam Drozdek. *Data Structures and Algorithms in C++.* 3rd Ed., Thomson.
3. A.Y. Langsam and A.M. Tanenbaum. *Data Structures Using C and C++.* Hall of India Pvt. Ltd., 2006.
4. W. Savitch. *Problem Solving with C++*, The OOP. 4th Ed., Pearson Education.

Title of the Course: Data Ware Housing & Data Mining
Category: PC
Course Code: 24BCOM23T
Branches: **Computer Science, AI&ML, AI&DA, Programming & Data Science**
Semester: II Semester

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	0	0	3

Course Objectives:

1. Learn the types of data to be mined and apply preprocessing methods on raw data.
2. To design data warehouses and techniques for mining frequent patterns, associations, and correlations.
3. To understand different classification algorithms and estimate the accuracy of algorithms.
4. To inculcate knowledge on different clustering algorithms.
5. To identify the various types of complex data and its applications.

Course Outcomes:

At the end of the course, the student will be able to

1. understand and apply the data preprocessing techniques.
2. design data warehouses and techniques for mining frequent patterns, associations, and correlations.
3. solve different classification problems and estimate the accuracy of classification algorithms.
4. understand and analyze different clustering techniques.
5. create various types of complex data such as spatial, text and multimedia.

Unit 1 INTRODUCTION: 12

Fundamentals of data mining, Data Mining Functionalities, Classification of Data Mining systems, Major issues in Data Mining.

Data Preprocessing: Needs Preprocessing the Data, Data Cleaning, Data Integration and Transformation, Data Reduction, Discretization and Concept Hierarchy Generation.

Unit 2 WARE HOUSING, MINING ASSOCIATION RULES IN LARGE DATABASES: 8

Data Warehouse and OLAP Technology for Data Mining Data Warehouse, Multidimensional Data Model, Data Warehouse Architecture, From Data Warehousing to Data Mining.

Basic Concepts and a Road Map, Efficient and Scalable Frequent Item set Mining Methods, Mining Various Kinds of Association Rules, From Association Mining to Correlation Analysis.

Unit 3 CLASSIFICATION AND PREDICTION 8

Issues Regarding Classification and Prediction, Classification by Decision Tree Induction, Bayesian Classification, Classification by Back propagation, Other Classification Methods, Prediction, Classifier Accuracy

Unit 4 CLUSTER ANALYSIS INTRODUCTION 10

Types of Data in Cluster Analysis, A Categorization of Major Clustering Methods, Partitioning Methods, Hierarchical Methods Density-Based Methods, Grid-Based Methods, Model-Based Clustering Methods, Outlier Analysis.

Unit 5 MINING COMPLEX TYPES OF DATA AND DATA MINING APPLICATIONS: 10

Mining Spatial Databases, Mining Multimedia Databases, Mining Text Databases, Mining theWorld Wide Web, Data Mining Applications.

Prescribed Textbooks:

1. Jiawei Han &Micheline Kamber. *Data Mining Concepts and Techniques.* 2nd Ed., 2006.

Reference Books:

1. Margaret H Dunham. *Data Mining Introductory and advanced topics.* Pearson Education.
2. Arun K Pujari. *Data Mining Techniques.* University Press.
3. Sam Anahory & Dennis Murray. *Data Warehousing in the Real World.* Pearson Edn Asia.
4. Paulraj Ponnaiah. *Data Warehousing Fundamentals.* Wiley Student Edition.
5. Ralph Kimball. *The Data Warehouse Life cycle Tool kit.* Wiley Student Edition

Title of the Course: Environmental Studies
Category: BS
Course Code: 24BCHE21T
Branches: **Computer Science, AI&ML, AI&DA, Programming & Data Science**
Semester: II Semester

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	0	0	3

Course Objectives: This course aims to raise awareness among students about the environment and the critical need to protect natural resources. It emphasizes the significance of ecosystems and biodiversity, while sensitizing students to the pollution challenges caused by everyday human activities. Additionally, the course equips students with the skills to identify and address social issues related to the environment.

Course Outcomes: At the end of the course, the student will be able to

1. explain how natural resources should be utilized with a focus on sustainability.
2. describe the need to protect ecosystems and biodiversity for future generations.
3. comprehend major pollution problems related to ecosystems.
4. apply quantitative reasoning skills in the proper utilization of goods and services.
5. analyze the interconnectedness of human dependence on Earth's ecosystems.

Unit 1 Multidisciplinary Nature of Environmental Studies 10

Definition, Scope and Importance – Need for Public Awareness. Natural resources: Renewable and non-renewable resources – Forest resources: Uses, deforestation– Water resources: Uses, floods, drought – Mineral resources: Uses, environmental effects of extracting mineral resources – Food resources: Impacts of overgrazing, problems with traditional agriculture, effects of modern agriculture – Land Resources: Land degradation, soil erosion Energy resources: Renewable and non-renewable energy resources.

Unit 2 Ecosystems, Biodiversity and its Conservation 10

Ecosystems: Producers, consumers and decomposers – Food chains, food webs and ecological pyramids – Introduction, types, characteristic features, structure and function of the following ecosystem: Forest ecosystem, lake ecosystem.
Biodiversity and Its Conservation: Definition – Value of biodiversity. Hot-spots of biodiversity – Threats to biodiversity – Conservation of biodiversity.

Unit 3 Environmental Pollution 8

Definition, Causes, effects and control measures of: Air Pollution, Water pollution, Soil pollution, Marine pollution, Noise pollution, Thermal pollution, nuclear hazards. Solid waste management: Causes, effects and control measures of urban and industrial wastes. e-waste: sources and health hazards. Need for e-waste management – recycling.

Unit 4 Social Issues and the Environment 10

Rain water harvesting, Environmental ethics: Issues and possible solutions – global warming, acid rain, ozone layer depletion – Environment Protection Act – Air (Prevention and Control of Pollution) Act – Water (Prevention and control of Pollution) Act – Wildlife Protection Act – Forest Conservation Act.

Unit 5 Human Population and the Environment 8

Population explosion – Family Welfare Programmes – Environment and human health – Value Education – HIV/AIDS – Role of information Technology in Environment and human health, Field work – Visit to a local area to document environmental assets

Prescribed Textbooks:

1. Kaushik, Anubha, and C.P. Kaushik. *Perspectives in Environmental Studies*. 6th Ed., New Age International Publishers, 2018.
2. Chawla, Shashi. *A Textbook of Environmental Studies*. McGraw Hill Education, 2017.

Reference Books:

1. Joseph, Benny. *Environmental Studies*. 3rd Ed., McGraw Hill Education, 2017.
2. Dhinakaran, A., and B. Sankaran. *A Textbook of Environmental Studies*. Himalaya Publishing House, 2017.
3. Basu, Mahua, and S. Xavier. *Fundamentals of Environmental Studies*. Cambridge University Press, 2017.
4. Bharucha, Erach. *Textbook of Environmental Studies for Undergraduate Courses*. University Press, 2013.
5. Tiwari, Vijay Kumar. *A Textbook of Environmental Studies*. Himalaya Publishing House, 2017.

Title of the Course: Python Programming Lab
Category: PC
Course Code: 24BCOM21L
Branches: **Computer Science, AI&ML, AI&DA, Programming & Data Science**
Semester: II Semester

Lecture Hours	Tutorial Hours	Practice Hours	Credits
0	0	4	2

Course Objectives:

1. To practice basics of python programming and basic control structures.
2. To practice python programming basic constructs like lists, dictionaries, sets and functions
3. To practice module design and usage of exception handling in python programming
4. To practice basics of object-oriented programming and elementary data structures.

Course Outcomes:

At the end of the course, the student will be able to

1. Apply basics of python programming
2. Write programs on the basic object-oriented programming in python language, handling of exceptions
3. Implement linear data structure in python programming
4. Develop and write programs for priority queues
5. Construct and write the implementation of binary search tree

List of Programs

Exercise 1: Install Python ecosystem and execute “Hello World” program.

Exercise 2: Practice

a) Python literals, variables, identifiers and data types
b) Python operators
c) Input and output statements.
d) Control statements

Exercise 3: Practice Python Programs on Numbers

a) Prime Numbers
b) Armstrong Numbers
c) Fibonacci Numbers and Series
d) Sum of squares for the first n natural numbers.
e) Reverse of a number

Exercise 4: Practice python programs on Various types of triangle patterns

Exercise 5: Implement python programs on functions, find factorial and Fibonacci number using recursion

Exercise 6: Practice python programs on lists, sets and dictionaries

Exercise 7: Practice any one python program on module design

Exercise 8: Practice python programs on string processing and exception handling

Exercise 9: Practice Python Programs

a) Write python program to implement encapsulation and abstraction
b) Write a python program to implement class variables and object variables

Exercise 10: Practice Python Programs

a) Write a python program to implement static variables and static methods.
b) Write a python program to implement super ()
c) Write a python program to implement types of inheritance

Exercise 11: Practice python programs

a) Write a python program to implement the method overloading and method overriding.
b) Write a python program to implement the abstract classes and interfaces.

Exercise 12: Implement python programs on

a) Stacks
b) Queues

Exercise 13: Implement Single linked list data structure.

Exercise 14: Implement priority queue data structure.

Exercise 15: Implement binary search tree data structure.

Prescribed Textbooks:

1. Charles Dierbach*, Introduction to Computer Science Using Python: A Computational Problem-Solving Focus*.
2. Rance D. Necaise, *Data Structures and Algorithms using Python*, Wiley Publications.

Reference Books:

1. R. Nageswara Rao. *Core Python Programming.* 2017 Ed., Dream Tech Press (Wiley India).
2. Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser. *Data Structures and Algorithms in Python.* Wiley Publications
3. Reema Thareja. *Python Programming using problem solving approach*, Oxford University press
4. John Zelle, Franklin, Beedle. *Python Programming: An Introduction to Computer Science.* 3rd Ed.
5. Allen Downey. *Think Python: How to think like a computer Scientist.* 2nd Ed., O'Reilly Publications
6. Bradley Miller, David L. Ranum, Franklin, Beedle. *Problem solving with algorithms and data structures using python.* independent publishers

Title of the Course: Object Oriented Programming using C++ Lab
Category: PC
Course Code: 24BCOM22L
Branches: **Computer Science, AI&ML, AI&DA, Programming & Data Science**
Semester: II Semester

Lecture Hours	Tutorial Hours	Practice Hours	Credits
0	0	4	2

Course Objectives:

1. To make the student learn an object-oriented way of solving problems.
2. To make the student write ADTs for all data structures.
3. To make the student learn different algorithm design techniques.

Course Outcomes:

At the end of the course, the student will be able to

1. Able to know about Object oriented programming concepts like Encapsulation, Constructors & Destructor, function overloading
2. Able to construct OOP concepts like Operator overloading, inheritance, and polymorphism
3. Able to use Abstract Data Types for data structures with templates
4. Able to design the different trees and their operations
5. Able to understand the implementation of Pattern Matching Algorithm

Week1:

a) Write a C++ program to implement the access control.
b) Write a C++ program to implement the static member function.
c) Write a C++ program to implement the parameter passing.

Week2:

a) Write a C++ program to implement the friend function.
b) Write a C++ program to implement the inline method.
c) Write a C++ program to implement dynamic memory allocation and deallocation.

Week 3:

a) Write a C++ program to implement the exception handling.
b) Write a C++ program to implement the constructor overloading.
c) Write a C++ program to implement the function overloading.

Week4:

a) Write a C++ program to implement the Operator overloading.
b) Write a C++ program to implement the simple inheritance.
c) Write a C++ program to implement multiple inheritance.

Week5:

a) Write a C++ program to implement the virtual function.

b) Write a C++ program to implement the abstract class.
c) Write a C++ program to implement the class template.

Week6: Write a C++ programs to implement the following using an array.
a) Stack ADT
b) Queue ADT

Week7: Write a C++ programs to implement the following using a singly linked list.
a) Stack ADT
b) Queue ADT

Week8: Write a C++ program to implement all the functions of a dictionary ADT using hashing.

Week9: Write a C++ program to perform the following operations on Binary Trees.
a) Insertion
b) Deletion
c) Searching

Week10: Write C++ programs to perform the traversals for the given binary tree.
a) Preorder
b) inorder
c) postorder

Week11: Write C++ programs for priority queue implementation using Heaps.
a) Min Heap Insertion
b) Min Heap Deletion
c) Max Heap Insertion
d) Max Heap Deletion

Week12: Write a C++ program to perform the following operations on Binary SearchTrees.
a) Insertion
b) Deletion
c) Searching

Week13: Write a C++ program to perform the following operations on B-Trees.
a) Insertion
b) Deletion
c) Searching

(Note: Use Class Templates for the above Data Structure Programs)

Prescribed Textbooks:

1. Adam Drozdek. *Data Structures and Algorithms in C++.* 3rd Ed., Thomson, 2007.
2. D.S. Malik. *Data Structures using C++.* Thomson.

Title of the Course: Operating Systems
Category: PC
Course Code: 24BCOM31T
Branches: **Computer Science, AI&ML, AI&DA, Programming & Data Science**
Semester: III Semester

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	0	0	3

Course Objectives: The goal is to provide a comprehensive understanding of operating system architecture and its functions, with a focus on gaining knowledge in process and thread synchronization, scheduling, paging, memory management techniques, and file system concepts. Additionally, the aim is to explore deadlock operations and gain insights into protection and security within operating systems.

Course Outcomes:
At the end of the course, the student will be able to
6. understand the fundamental concepts of operating systems
7. explain the concepts of process management and threads.
8. describe efficient utilization of memory Management and file system techniques.
9. analyze and investigate the impacts of deadlocks.
10. understand security and Protection mechanisms in operating systems

Unit 1 Operating Systems Overview 9
Introduction, what operating systems do? Computer system Organization & architecture, Operating system operations.
Systems structures: Operating system services, systems calls, types of System calls.

Unit 2 Process Management 12
Process concepts; process Scheduling, operations on process, Process Scheduling Basic Concepts, Scheduling Criteria, scheduling algorithms, IPC, communication in Client-Server systems.

Unit 3 Memory Management 12
Swapping, contiguous memory allocation, paging, structure of the page table, segmentation, Demand paging, page-replacement algorithms
File System: File Concept, access methods, file allocation methods in disk Space, disk scheduling

Unit 4 Principles of Deadlock 9
System model, deadlock characterization, deadlock prevention, detection and avoidance, recovery form deadlock.

Unit 5 I/O Systems 9
I/O Hardware, Transforming I/O requests to hardware operations.

Protection: Protection, Goals of Protection, Principles of Protection.
Security: The Security problem, program threats, system and network threats, firewalls to protect systems and networks.

Prescribed Textbooks:

1. Abraham Silberchatz, Peter B. Galvin, Greg Gagne, *Operating System Concepts*. 10th Ed., John Wiley.

Reference Books:

1. Andrew S. Tanenbaum, *Modern Operating Systems*. 2nd Ed., PHI.
2. William Stallings, *Operating Systems: Internals and Design Principles*. 6th Ed. Pearson Education, 2009.B.L.Stuart, *Principles of Operating Systems*. Cengage learning, India Edition.
3. D. M. Dhamdhere, *Operating Systems a Concept based Approach*. 2nd Ed., TMH.

Title of the Course: Software Engineering
Category: PC
Course Code: 24BCOM32T
Branches: **Computer Science**
Semester: III Semester

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	0	0	3

Course Objectives: The software life cycle is a structured process for developing, deploying, and maintaining software through stages like requirement gathering, design, implementation, testing, deployment, and maintenance. Process models such as Waterfall, Agile, and Spiral provide frameworks to manage these stages based on project needs. Gathering requirements ensures alignment with user needs, while effective designs at various levels create scalable systems. Testing strategies like unit, integration, and acceptance testing ensure software quality. Development and maintenance focus on implementation and updates to ensure reliability and longevity.

Course Outcomes:
At the end of the course, the student will be able to
1. understand the software process models importance.
2. analyze software requirements through elicitation, specification, and validation techniques based on requirements engineering principles..
3. understand the software architecture with various design approaches..
4. design user interfaces by applying established principles, analysis techniques, and structured design steps..
5. analyze the maintenance of a software project.

Unit 1 Software and Software Engineering 12

Software and Software Engineering: The Nature of Software, Software Engineering, Software Process, Software Engineering Practice, Software Myths.
Process Models: A Generic Process Model (framework), Process Assessment and Improvement, Prescriptive Process Models, The Unified Process.

Unit 2 8

Understanding Requirements: Requirements Engineering, Software Requirement Specification, Eliciting requirements, and Validating Requirements.
System Modeling: Context models, Behavioral models, Data models, Object models, Structured Methods.

Unit 3 8

Design Concepts: Design Concepts. Architectural Design: Software Architecture, Architectural Styles, Architectural Design, Assessing Alternative Architectural Designs, Design Evaluation.

Unit 4 10

User Interface Design: The Golden Rules, User Interface Analysis and Design, Interface Design Steps. **Testing:** Unit Testing, Black-Box Testing, White-Box Testing, Integration Testing, System Testing.

Unit 5 10

Project Management: The Management Spectrum, People, Product, Process, Project, Project Planning, Project Scheduling.

Risk Management: Risk Identification, Risk Projection, Risk Refinement, Risk Mitigation, Monitoring, and Management, The RMMM Plan.

Prescribed Textbooks:

1. Roger Pressman, *Software Engineering: A Practitioner's Approach*. 9th Ed., Mc Graw Hill International Edition, 2023.
2. Ian Sommerville, *Software Engineering*. 9th Ed., Pearson education, 2011.

Reference Books:

1. Waman S Jawadekar, *Software Engineering: A Primer.* T*ata* McGraw-Hill, 2004.
2. Rajib Mall*, Fundamentals of Software Engineering*. 5th Ed., PHI, 2005.
3. Pankaj Jalote, *Software Engineering: A Precise Approach*. Wiley India.
4. Deepak Jain*, Software Engineering: Principles and Practices*. Oxford University Press, 2008.

Title of the Course: Discrete Mathematics
Category: BS
Course Code: 24BMAT31T
Branches: **Computer Science, AI&ML, AI&DA, Programming & Data Science**
Semester: III Semester

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	0	0	3

Course Objectives: The course aims to develop the foundational understanding of discrete mathematics essential for computer science and engineering students. It enables students to formulate and analyze compound propositions using truth tables and logical equivalences, integrate concepts from sets and functions to solve engineering problems, and derive closed-form solutions of linear recurrence relations using various methods. The course also aims to build awareness of the fundamental concepts of graph theory and trees, along with their applications in computing and problem-solving contexts.

Course Outcomes:
At the end of the course, the student will be able to
1. describe the logical connectives in inference theory
2. explain relations, functions in computer science and engineering.
3. solve the various types of recurrence relations in engineering
4. apply graph theory concepts in network analysis
5. apply the properties of trees in minimization of the path distance.

Unit 1 Mathematical Logic 10
Introduction, Statements and Notation, Connectives-negation, conjunction, disjunction, conditional and bi-conditional, Well-formed formulas, Tautology, Duality law, Equivalence, Implication, Normal Forms, Inference Theory of Statement Calculus.

Unit 2 Relations and functions 10
Relations and functions: Operations on Relations, Properties of binary relations in a set, relation matrix and the graph of a relation, partition and covering of a set, equivalence relations, compatibility relations, partial ordering, Hasse diagram, functions - composition of functions, inverse functions and recursive functions.

Unit 3 Recurrence Relations 8
Generating Functions of Sequences, Calculating Coefficients of Generating Functions and Recurrence Relations-Solving homogeneous and non-homogeneous Recurrence Relations by Substitution and Generating functions and Characteristic roots method.

Unit 4 Graph theory 8
Definitions, finite and infinite graphs, incidence and degree, isolated pendant vertices, isomorphism, sub

graphs, walk, path and circuit, connected and disconnected graphs, components, Euler graphs, Euler graph theorem, operations on graphs, arbitrarily traceable Euler graphs, Hamiltonian paths and circuits, travelling salesman problem.

8

Unit 5 Trees

Some properties of trees, pendant vertices, distance and centres, rooted and binary trees, spanning trees, fundamental circuit, shortest spanning trees, Kruskal's algorithm.

Prescribed Textbooks:

1. J.P. Tremblay and R. Manohar. *Discrete Mathematical Structures with Applications to Computer Science*. Tata McGraw Hill, 2002.
2. K. H. Rosen. *Discrete Mathematics and its Applications with Combinatorics and Graph Theory*. 7th Ed., McGraw Hill Education (India) Private Limited.

Reference Books:

1. J. L. Mott, A. Kandel and T. P. Baker, *Discrete Mathematics for Computer Scientists & Mathematicians*. 2nd Ed., Pearson Education.
2. N. Deo, *Graph Theory with Applications to Engineering and Computer Science*. Prentice Hall of India, 2006.

Title of the Course: Object Oriented Modeling and Design
Category: PC
Course Code: 24BCOM33T
Branch/es: **Computer Science, AI&ML, AI&DA, Programming & Data Science**
Semester: III Semester

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	0	0	3

Course Objectives: The course aims to help students understand the role and significance of modeling in software development. It aims to teach the principles of object-oriented modeling and how to apply them using UML. The course aims to cover the various building blocks and architecture of UML to enable effective software system modeling. It aims to develop the ability to design and interpret UML structural diagrams such as class, object, and component diagrams. Additionally, it aims to provide practical knowledge of behavioural modeling through use case, sequence, collaboration, and state chart diagrams. The course also aims to foster understanding of deployment and architectural diagrams and their importance in system implementation. Ultimately, it aims to equip students with the skills to apply UML modeling techniques to real-world software problems, bridging the gap between system requirements and implementation.

Course Outcomes:
At the end of the course, the student will be able to

1. Understand the fundamental concepts of modeling, its importance in software engineering, and the principles of object-oriented modeling.
2. Analyze and construct detailed structural models using advanced class features, relationships, interfaces, roles, and packages.
3. Identify and describe system functionality using use cases and actors.
4. Demonstrate understanding of dynamic system behavior using activity diagrams, state machines, and state chart diagrams.
5. Design and interpret component diagrams to illustrate system components and their interfaces.

Unit 1 Introduction to UML 12

Importance of modeling, principles of modeling, object-oriented modeling, An overview of UML, Building Blocks of UML, Architecture, SDLC, Classes – Terms and Concepts, Common Modeling Techniques, Relationships.

Unit 2 Structural Modeling 8

Advanced classes, advanced relationships, Interfaces, Types and Roles, Packages, Instances**, Class Diagrams** – Terms, concepts for Class Diagram.

Unit 3 Basic Behavioural Modeling 8

Use cases – Terms and Concepts, **Use case Diagrams** – Terms and Concepts, **Sequence Diagrams** – Terms and Concepts, Collaboration Diagrams – Terms and Concepts.

Unit 4 Advanced Behavioral Modeling 10

Activity Diagrams – Terms and Concepts, Events and Signals, State Machines, Processes and Threads, Time and Space, State Chart Diagrams - Terms and Concepts.

Unit 5 Architectural Modeling 10

Component Diagrams – Terms and Concepts, Deployment Diagrams – Terms and Concepts.

Prescribed Textbooks:

1. Grady Booch, James Ram Baugh and Ivar Jacobson, *The Unified Modeling Language User Guide*. Pearson Education, 1999.

Reference Books:

1. John W. Satzinger, Robert B. Jackson and Stephen D. Burd, *Object-Oriented Analysis and Design with the Unified Process*. Cengage Learning, 2004.
2. Hans-Erik Eriksson, Magnus Penker, Brian Lyons and David Fado, *UML 2: Toolkit*. Wiley India Pvt. Ltd., 2004.

Title of the Course: Computer Networks
Category: PC
Course Code: 24BCOM34T
Branches: **Computer Science, AI&ML, AI&DA, Programming & Data Science**
Semester: III Semester

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	0	0	3

Course Objectives: This course aims to provide an understanding of protocol layering and physical-level communication, along with the ability to analyze network performance. It covers the functions of the network layer and various routing protocols, while also familiarizing students with the transport layer's roles and protocols like TCP and UDP. Additionally, it explores the working of application layer protocols such as HTTP, FTP, and DNS, enabling a deeper insight into how networks support seamless data communication.

Course Outcomes:
At the end of the course, the student will be able to

1. classify the different aspects of networks, protocols and network design models.
2. examine various Data Link layer design issues and Data Link protocols.
3. analyze, Compare and select appropriate routing algorithms for a network.
4. examine the various end to end protocols helps in analyzing and interpreting the quality of networks.
5. analyze the various application protocols over internet.

Unit 1 Introduction to Networks and Physical Layer 9
Network hardware, Network software, OSI, TCP/IP Reference models, Example Networks: ARPANET, Internet. Physical Layer: Guided Transmission media: twisted pairs, coaxial cable, fiber optics, Wireless transmission.

Unit 2 Data Link Layer 9
Data link layer: Design issues, framing.
Elementary data link protocols: simplex protocol, Sliding Window protocols- A one-bit sliding window protocol.
Medium Access sub layer: The channel allocation problem, Multiple access protocols: ALOHA, Carrier sense multiple access protocols.

Unit 3 Network Layer 9
Design issues, Routing algorithms: shortest path routing, Flooding, Hierarchical routing, Broadcast, Multicast, distance vector routing, Quality of Service, Internetworking, The Network layer in the internet.

Unit 4 Transport Layer 9
Transport Services, Elements of Transport protocols, Connection management, TCP and UDP protocols.

Unit 5 Application Layer 9

Domain name system, SNMP, Electronic Mail; the World WEB, HTTP.

Prescribed Textbooks:

1. Andrew S. Tanenbaum, David J. Wetherall, *Computer Networks*. 5th Ed., Pearson New International Edition, 2016.

Reference Books:

1. William Stallings, *Data and Computer Communication*. 8th Ed., Pearson, PHI, 2013.
2. Douglas Comer*, Internetworking with TCP/IP*, 6th Ed., PHI, 2015.
3. Behrouz A. Forouzan, *Data Communication and Networking*. 4th Ed., McGraw- Hill, 2017.

Title of the Course: Essence of Indian Traditional Knowledge
Category: MC
Course Code: 24ENG31T
Branches: **Computer Science, AI&ML, AI&DA, Programming & Data Science**
Semester: III Semester

Lecture Hours	Tutorial Hours	Practice Hours	Credits
2	0	0	0

Course Objectives: The objective is to learn the basic principles of thought process, reasoning, and inference, sustainability at the core of Indian Traditional Knowledge Systems that connect society and nature. It also aims to understand the holistic lifestyle of Yogic science and the wisdom capsules found in Sanskrit literature, especially in the context of modern society marked by rapid technological advancements and societal disruptions. Furthermore, it emphasizes understanding the Indian Knowledge System, the Indian perspective on the modern scientific worldview, and the foundational principles of Yoga and holistic healthcare. Additionally, it includes exploring Indian philosophical traditions, linguistic traditions, and artistic heritage.

Course Outcomes:
At the end of the course, the student will be able to

1. Understand the hidden scientific outlook and imbibed human values in the Indian way of life.
2. Understand the essence of Vedas and their value
3. Establish connection between Indian knowledge system and Modern science.
4. Explain basics of Indian artistic, linguistic and philosophical tradition
5. Explain basics of philosophical tradition.

Unit 1 Indian Tradition 9

Fundamental unity of India, India's heroic role in world civilization, The Indian way of life, Introduction to Indian tradition, The Scientific Outlook and Human Values.

Unit 2 Basic structure of Indian Knowledge System 9

Indian Traditional Scriptures, Exposure to 4-Vedas (the Rigveda, the Yajurveda, the Samaveda and the Atharvaveda) , 4-Upvedas (Ayurveda, Dhanurveda, Gandharvaveda, Sthapatya etc.), 6-Vedangas (Shiksha, Kalp, Nirukta, Vyakaran, Jyotish), 6-Upangas (Dharmashastra, Meemansa, Puranas, Tarkashastra/Logic) etc.

Unit 3 Indian Knowledge System and Modern Science 9

Relevance of Science and Spirituality, Science and Technology in Ancient India, Superior intelligence of Indian sages and scientists.
Indian Traditional Health Care: Importance and Practice of Yoga, Pranayama and other prevailing health care techniques..

Unit 4 Indian Artistic and Linguistic Tradition 9

Indian Artistic Tradition: Introduction and overview of significant art forms in ancient India such as painting, sculpture, Civil Engineering, Architecture, Music, Dance, Literature etc.

Indian Linguistic Tradition: Ancient Indian languages and literary Heritages, Phonology, Morphology, Syntax and Semantics..

Unit 5 Indian Philosophical Tradition 9

Indian Philosophical Tradition: (Sarvadarshan)- Nyaya, Viaisheshiika, Sankhya, Yoga, Meemansa, Brief understanding of Philosophy of Charvaka, Bhagwan Mahaveer Jain, Bhagwan Buddha, Kabeer, Guru Nanak Dev and other eminent ancient Indian Philosophers.

Activities: Activities will consist of one assignment on each module, group discussions, presentations, case study on various topics based on above curriculum.

Prescribed Textbooks:

1. L. H. Ajwani, *Immortal India*. Vora & Co. Publishers, 1997.
2. V. V. Raman, *Glimpses of Indian Heritage*. Popular Prakashan,1993

Reference Books:

1. V. Sivaramakrishnan, *Cultural Heritage of India- Course Material*. Bharatiya Vidya Bhavan, Mumbai, 5 th Ed., 2014.
2. F. Capra, *Tao of Physics*. Shambhala, 2010.
3. S. C. Chaterjee and D. M. Datta, *An Introduction to Indian Philosophy*. University of Calcutta, 1984.
4. Krishna Chaitanya, *Arts of India*. Abhinav Publications, 1987.
5. V.N. Jha, *Language*, Thought and Reality.

Title of the Course: Software Engineering Lab
Category: PC
Course Code: 24BCOM32L
Branches: **Computer Science**
Semester: III Semester

Lecture Hours	Tutorial Hours	Practice Hours	Credits
0	0	4	2

Course Objectives: The course aims to apply software engineering principles to real world scenarios and projects. To Enable students to understand and document both functional and non-functional requirements various system modeling techniques effectively. To Develop skills in making critical architectural decisions that influence the overall system structure. To Apply verification, validation, and testing concepts to real-world software projects through hands-on exercises and projects. To Apply project management, planning, scheduling, risk management, and cost estimation concepts to real-world software projects.

Course Outcomes:
At the end of the course, the student will be able to
1. Comprehend various requirements needed to develop software.
2. Apply various process models for developing software
3. Apply different modular decomposition styles for developing software architecture

Design and Develop following applications:
1. Online Railway Reservation System
2. Payroll processing application
3. Inventory system.
4. Automating the banking process.
5. Software for game
6. Library management system
7. Create a dictionary
8. Text editor
9. Telephone directory
10. Create an E-Book of your choice

Prescribed Textbooks:

1. Rogers Pressman, *Software Engineering: A Practitioner's Approach*. 7th Ed., Mc. Graw Hill International Edition, 2009.

Reference Books:

1. Ian Sommerville, *Software Engineering*. 9th Ed., Pearson education, 2011.
2. Waman S Jawadekar, *Software Engineering: A Primer.* T*ata* McGraw-Hill, 2004.
3. Pankaj Jalote, *Software Engineering: A Precise Approach*. 2nd Ed., Wiley India.

4. Deepak Jain*, Software Engineering: Principles and Practices*. Oxford University Press, 2008.

Title of the Course: UML Lab
Category: PC
Course Code: 24BCOM33L
Branches: **Computer Science, AI&ML, AI&DA, Programming & Data Science**
Semester: III Semester

Lecture Hours	Tutorial Hours	Practice Hours	Credits
0	0	4	2

Course Objectives: The course aims to help students understand the core principles and significance of modeling in software development. It aims to enable students to create structural diagrams, including class diagrams, to represent the static aspects of a system. The course aims to develop students' ability to create use case and sequence diagrams to effectively model system interactions and behavior. It also aims to teach the design of advanced behavioral models using activity diagrams to capture dynamic workflows and processes. Additionally, it aims to guide students in designing component and deployment diagrams to illustrate the architectural and physical deployment aspects of software systems.

Course Outcomes:
At the end of the course, the student will be able to
1. Creating UML diagrams using standard UML modeling tools
2. Design class diagrams with complex relationships, interfaces, and packages.
3. Apply skills in representing dynamic behaviors of systems through use case, sequence, and collaboration diagrams.
4. Apply activity and state chart diagrams to model workflows and system states
5. Design component and deployment diagrams

Select at least 3 from the following and design UML Diagrams:

1. Online Railway Reservation System
2. Library management system
3. Hospital Management System
4. ATM Model
5. College Management System

Prescribed Textbooks:

1. Grady Booch, James Rumbaugh, Ivar Jacobson, *The Unified Modeling Language User Guide*. 2nd Ed., Pearson Education, 2007.

Reference Books:

1. Meilir Page-Jones, *Fundamentals of Object Oriented Design in UML*. Pearson Education, 1sr Ed., 2006.
2. John W. Satzinger, Robert B. Jackson and Stephen D. Burd, *Object-Oriented Analysis and Design with*

the Unified Process. Cengage Learning, 2004.

Title of the Course: Web Technologies
Category: PC
Course Code: 24BCOM41T
Branches: **Computer Science, AI&ML, AI&DA, Programming & Data Science**
Semester: IV Semester

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	0	0	3

Course Objectives: Developing effective web pages involves a comprehensive understanding of HTML, CSS, JavaScript, and XML. HTML provides the foundational structure, enabling the creation of tables and form controls to organize content and facilitate user interactions. CSS is employed to design and style these web pages, ensuring a responsive and visually appealing layout. JavaScript is utilized to validate form inputs, enhancing user experience by providing immediate feedback and preventing erroneous data submissions. Additionally, XML can be used to structure and transport data, allowing for the design of web pages that efficiently handle and display complex information. By integrating these technologies, developers can create dynamic, well-structured, and user-friendly web applications.

Course Outcomes:
At the end of the course, the student will be able to
1. apply the fundamental HTML markups when designing web pages.
2. apply Tables and forms controls to design web pages.
3. apply cascading style sheets to design web pages
4. apply JavaScript concepts in designing web pages.
5. apply XML features for developing web pages.

Unit 1 10
Introduction to HTML, Tags and Elements, Basic Text Formatting, Understanding Block and Inline Elements, Text Processing tags, Links and Navigation: Basic Links, Understanding Directories and Directory Structures, Understanding URLs, Creating In-Page Links with the <a> Element.

Unit 2 6
Tables: Introducing Tables, Basic Table Elements and Attributes, adding a Caption to a Table, Grouping Sections of a Table, Forms: Introducing Forms, creating a Form with the <form> Element, Form Controls, Creating Labels for Controls and the <label> Element, Frames.

Unit 3 8
Cascading Style Sheets: Introducing CSS, Types of CSS, Text Formatting, Styling Text, Backgrounds, Lists.

Unit 4 Introduction to JavaScript 10

Introduction to JavaScript, Declaring Java Script Variables, Basics of JavaScript, Control Structures, Pop Up Boxes, Functions, Arrays, Events, Objects.

Unit 5 Introduction to XML 10

XML: Introduction, Document Type Definition, XML Schema, Document Object Model, Presenting XML (XSL), XML Parsers: DOM and SAX.

Prescribed Textbooks:

1. Rob Larsen, *Beginning HTML and CSS* (Wrox Programmer to Programmer). Wiley, 2013.
2. Jon Duckett, *Java Script and JQuery: Interactive Front-End Web Development*. Wiley Publications, 2014.

Reference Books:

1. Erik T. Ray, Learning XML O'REILLY Publications, 2nd Edition, 2003.
2. Jon Duckett, *Web Design with HTML CSS JavaScript and JQuery Set*,. Wiley Publications, 2014.

Title of the Course: Software Project Management
Category: PC
Course Code: 24BCOM42T
Branches: **Computer Science**
Semester: IV Semester

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	0	0	3

Course Objectives: Gaining a comprehensive understanding of software management fundamentals involves studying the Software Development Life Cycle (SDLC) phases and the role of various artifacts. This includes understanding key software management disciplines and acquiring the skills necessary to effectively gather, analyze, and use data. Additionally, to equip students or professionals with the knowledge and skills to manage and lead software development teams effectively.

Course Outcomes:
At the end of the course, the student will be able to

1. understand the conventional software Management and Software Economics factors that affect the development of a software project.
2. use artifacts in Software Development lifecycle
3. apply software management disciplines effectively in diverse project environments.
4. analyze case studies to identify successful practices and opportunities for process optimization
5. apply Leadership Concepts in Software Project Management.

Unit 1 12
FUNDAMENTALS: Conventional software management - Evolution of software economics - Improving software economics - Conventional versus modern software project management.

Unit 2 8
Life cycle phases: Engineering and production stages, inception, Elaboration, construction, transition phases. Artifacts of the process: The artifact sets, Management artifacts, Engineering artifacts, pragmatic artifacts.

Unit 3 8
SOFTWARE MANAGEMENT DISCIPLINES: Iterative process plans - Organization and responsibilities - Process automation - Process control and process instrumentation - Tailoring the process.

Unit 4 10
MANAGED AND OPTIMIZED PROCESS: Data gathering and analysis - Principles of data gathering - Data gathering process - Software measures - Data analysis - Managing software quality - Defect prevention

Unit 5 10

SOFTWARE DEVELOPMENT: Software Development Team - Three Vital Aspects of Software Project Management -The Team - Meaning of Leadership - Communicating in Harmony - Personality traits - Project Organizations

Prescribed Textbooks:

1. Walker Royce, *Software Project Management: A Unified Framework*. 13th Ed., Pearson Education, 2002.
2. Watts S. Humphrey, *Managing the Software Process*. 1st Ed., Addison Wesley, 1989.

Reference Books:

1. Ramesh Gopalaswamy, *Managing Global Software Projects*. 1st Ed., Tata McGraw Hill, 2005.
2. Bob Hughes and Mike Cotterell, *Software Project Management*. 3rd Ed., Tata McGraw Hill, 2009.

Title of the Course: Design and Analysis of Algorithms
Category: PC
Course Code: 24BCOM43T
Branches: **Computer Science, AI&ML, AI&DA, Programming & Data Science**
Semester: IV Semester

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	0	0	3

Course Objectives: The objective of the course Design and Analysis of Algorithms is to introduce fundamental algorithm design techniques such as divide and conquer, greedy methods, and dynamic programming. It aims to develop students' ability to analyze the efficiency of algorithms in terms of time and space complexity. The course encourages the application of algorithms to solve real-world computational problems, particularly those relevant to Artificial Intelligence and Machine Learning. Additionally, it fosters strong problem-solving skills through the design and evaluation of optimal and efficient algorithms.

Course Outcomes:
At the end of the course, the student will be able to

1. understand the performance of algorithms in terms of time and space complexity
2. apply divide and conquer to solve searching and sorting problems and greedy method to solve job sequencing with deadline.
3. analyze dynamic programming technique to solve knapsack, shortest path, travelling sales person problems.
4. understand and solve different applications of backtracking, and branch and bound techniques
5. understand and remember concepts of computational theory.

Unit 1 12

Algorithm, Pseudo Code for algorithms, performance analysis-Space complexity, Time Complexity, Asymptotic Notation-Big Oh Notation, Omega Notation, Theta notation and Little Oh notation, Sets-Disjoint set operations, Union and Find algorithms.

Unit 2 8

Divide and Conquer: General Method, applications- Binary Search, Quick sort, Merge Sort.
Greedy Method: General Method, Prim's algorithm, Kruskal's algorithm, Single Source Shortest Paths-Dijkstra's algorithm

Unit 3 8

Dynamic Programming: General Method, applications-, Optimal Binary search trees, 0/1 Knapsack, All pairs shortest path: Floyd Warshal's, The Travelling person problem.

Unit 4 10

Backtracking: General Method, applications- N- queen problem, sum of subsets, graph coloring, Hamiltonian cycles.

Branch and Bound: General Method, applications, 0/1 knapsack problem-LC Branch and Bound Solution, FIFO Branch and Bound solution.

Unit 5 10

The classes: P-Class, NP-Class, Non-deterministic algorithms, NP-Hard Class and NP Complete Class, Cook's Theorem.

Prescribed Textbooks:

1. Ellis Horowitz, Sartaj Sahni and S. Rajasekaram, *Fundamentals of Computer Algorithms*. 2nd Ed., Universities Press.

Reference Books:

1. R. C. T. Lee, S. S. Tseng, R. C. Chang and Y. T. Tsai, *Introduction to Design and analysis of Algorithms*: *A strategic approach*. McGraw Hill, 2005.
2. Aho, Ullman and Hopcroft, *Design and Analysis of algorithms. Pearson Education*, 2004.
3. M.T. Goodrich and R. Tomassia, *Algorithm Design: Foundations, Analysis and Internet Example*. John Wiley and sons.

Title of the Course: Numerical Methods
Category: BS
Course Code: 24BMAT41T
Branches: **Computer Science, AI&ML, AI&DA, Programming & Data Science**
Semester: IV Semester

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	0	0	3

Course Objectives: The objective of the Numerical Methods course is to introduce students to numerical techniques for solving algebraic and transcendental equations, and to familiarize them with interpolation methods for estimating values between known data points. The course also aims to explain various numerical approaches for evaluating definite integrals and solving ordinary differential equations. Additionally, it describes the applications of numerical methods in solving partial differential equations, equipping students with practical tools for computational problem-solving in scientific and engineering contexts.

Course Outcomes:
At the end of the course, the student will be able to
1. solve algebraic and transcendental equations using numerical methods .
2. solve the simultaneous equations using numerical methods .
3. apply interpolation techniques to solve real-world problems by estimating intermediate values from discrete data points.
4. evaluate differentiation and integration of a function using numerical technique.
5. solve the ordinary differential equations using numerical methods .

Unit 1 Solutions of algebraic and transcendental equations 12
Bisection method - Regula-Falsi method - The Iteration Method-Newton-Raphson method.

Unit 2 Numerical solutions of simultaneous equations and Eigen value problems 10
Introduction, Gaussian elimination method, Gauss-Jordan elimination method. Iterative methods: Gauss Jacobi iteration method, Gauss-Seidel iteration method. Power method for determining the largest eigen values.

Unit 3 Interpolation and Curve fitting 8
Interpolation: Finite differences - forward differences and backward differences - Newton's forward interpolation formula - Newton's backward interpolation formula - Lagrange's interpolation formula.
Curve fitting: Fitting a straight line-second degree parabola-Exponential curve –power curve by the method of least squares.

Unit 4 Numerical Differentiation and Numerical integration 8

Numerical Differentiation, Numerical integration - Trapezoidal rule - Simpson's 1/3rd and Simpson's 3/8 rules.

Unit 5 Numerical solutions of ordinary differential equations of first order 10

Picard's method -Taylor's series- Euler's method - Modified Euler's method - Runge-Kutta method of fourth order

Prescribed Textbooks:

1. B. S. Grewal, *Higher Engineering Mathematics*. 44th Ed., Khanna Publishers, 2017.
2. R. K. Jain, S. R. K. Iyengar, *Advanced Engineering Mathematics*. 5th Ed., Narosa Publishers, 2016.

Reference Books:

1. P. Kandasamy, K. Thilagavathi, K. Gunavathi, *Numerical Methods*. S. Chand & Company, 2nd Ed., Reprint 2012.
2. N.P. Bali and Manish Goyal, *A text book of Engineering Mathematics*. Laxmi Publications, Reprint, 2010.
3. T. Veerarajan, *Engineering Mathematics*. Tata McGraw-Hill, New Delhi, 2008.

Title of the Course: Internet of Things
Category: PC
Course Code: 24BCOM44T
Branches: **Computer Science, AI&ML, AI&DA, Programming & Data Science**
Semester: IV Semester

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	0	0	3

Course Objectives: The course aims to provide a solid understanding of IoT fundamentals, including connectivity, data exchange, and automation. It covers IoT architecture and protocols, focusing on device communication within networks. Students will gain hands-on experience by building embedded systems with Raspberry Pi to design IoT solutions. The course also explores real-world IoT applications in industries like healthcare, agriculture, and smart cities, bridging theory with practical implementation.

Course Outcomes: At the end of the course, the student will be able to

1. Describe the physical and logical design of IoT systems, enabling technologies, levels of deployment, templates.
2. Develop Web services to access/control IoT devices.
3. Design a portable IoT using Raspberry Pi.
4. Deploy an IoT application and connect to the cloud.
5. Analyze applications of IoT in real time scenario.

Unit 1 Introduction to IOT **10**

Internet of Things - Physical Design- Logical Design- IoT Enabling Technologies - IoT Levels & Deployment Templates - Domain Specific IoTs.

Unit 2 IOT Architecture **10**

M2M high-level ETSI architecture - IETF architecture for IoT - OGC architecture - IoT reference model - Domain model - Information model - Functional model - Communication model - IoT Reference architecture.

Unit 3 IOT Protocols **8**

Protocol Standardization for IoT – Efforts – M2M and WSN Protocols – SCADA and RFID Protocols – Unified Data Standards – Protocols – IEEE 802.15.4 – BACNet Protocol – Modbus– Zigbee Architecture – Network layer- 6LowPAN - CoAP – Security.

Unit 4 Building IOT with Raspberry PI **10**

Building IOT with RASPBERRY PI- IoT Systems - Logical Design using Python – IoT Physical Devices & Endpoints - IoT Device -Building blocks

Unit 5 Case Studies and Real-World Applications 8

Real world design constraints - Applications - Asset management, Industrial automation, Smart grid, Commercial building automation, Smart cities - Participatory sensing - Data Analytics for IoT – Software & Management Tools for IoT Cloud Storage Models & Communication APIs - Cloud for IoT - Amazon Web Services for IoT.

Prescribed Textbooks:

1. Arshdeep Bahga, Vijay Madisetti*, Internet of Things: A hands-on approach*. Universities Press, 2015.

Reference Books:

1. Dieter Uckelmann, Mark Harrison, Michahelles, Florian (Eds), *Architecting the Internet of Things*. Springer, 2011
2. Honbo Zhou, *The Internet of Things in the Cloud: A Middleware Perspective*, CRC Press, 2012
3. Jan Holler, Vlasios Tsiatsis , Catherine Mulligan, Stamatis Karnouskos, Stefan Avesand and David Boyle, From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence", Elsevier, 2014.
4. Olivier Hersent, David Boswarthick, Omar Elloumi, *The Internet of Things – Key applications and Protocols*. Wiley, 2012.

Title of the Course: Constitution of India
Category: MC
Course Code: 24BENG41T
Branches: **Computer Science, AI&ML, AI&DA, Programming & Data Science**
Semester: IV Semester

Lecture Hours	Tutorial Hours	Practice Hours	Credits
2	0	0	0

Course Objectives: The course aims to provide an understanding of the importance of the Constitution and the structure of the executive, legislature, and judiciary. It explores the philosophy behind fundamental rights and duties, and the autonomous nature of constitutional bodies such as the Supreme Court, High Court, CAG, and Election Commission. The course also highlights the financial and administrative relations between the Union and State governments. Overall, it fosters awareness of constitutional governance in India.

Course Outcomes: At the end of the course, the student will be able to

1. Understand the historical background of the constitution making and its importance for building a democratic India.
2. Understand the functioning of three wings of the government, i.e., executive, legislative and judiciary
3. Understand the value of the fundamental rights and duties for becoming good citizens of India
4. Understand the decentralization of power between union, state and local self-government
5. Understand the operation of constitutional institutions like CAG, Election Commission and UPSC for sustaining democracy

Unit 1 **10**

IntroductiontoIndianConstitution:Constitution,meaningoftheterm,IndianConstitution-Sourcesand Constitutional history, Features Citizenship, Preamble, Fundamental Rights and Duties, Directive Principles of State Policy.

Unit 2 **10**

Union Government and its Administration Structure of the Indian Union: Federalism, Centre – State relationship, President: Role, power and position, PM and Council of ministers, Cabinet and Central Secretariat, LokSabha, Rajya Sabha, The Supreme Court and High Court: Powers and Functions.

Unit 3 **8**

State Government and its Administration – Governor – Role and Position – CM and Council of ministers, State Secretariat: Organization, Structure and Functions

Unit 4 **10**

Local Administration – District Administration Head – Role and Importance, Municipalities – Mayor and role of Elected Representative – CEO of Municipal Corporation Panchayati Raj: Functions PRI: Zila Panchayat, Elected officials and their roles, CEO Zila Panchayat: Block level Organization all Hierarchy - (Different departments), Village level – Role of Elected and Appointed officials – Importance of grass root democracy.

Unit 5 **8**

Election Commission: Election Commission – Role of Chief Election Commissioner and Election Commissionerate State Election Commission: Functions of Commissions for the welfare of SC/ST/OBC and women.

Prescribed Textbooks:

1. Durga Das Basu, *Introduction to the Constitution of India*. Prentice-Hall of India Pvt. Ltd. New Delhi
2. Subash Kashyap, *Indian Constitution*. National Book Trust

Reference Books:

1. J. A. Siwach, *Dynamics of Indian Government & Politics*
2. D. C. Gupta, *Indian Government and Politics*
3. M.V. Pylee, *India's Constitution*

Title of the Course: Web Technologies Lab
Category: PC
Course Code: 24BCOM41L
Branches: **Computer Science, AI&ML, AI&DA, Programming & Data Science**
Semester: II Semester

Lecture Hours	Tutorial Hours	Practice Hours	Credits
0	0	4	2

Course Objectives: Developing effective web pages involves a comprehensive understanding of HTML, CSS, JavaScript, and XML. HTML provides the foundational structure, enabling the creation of tables and form controls to organize content and facilitate user interactions. CSS is employed to design and style these web pages, ensuring a responsive and visually appealing layout. JavaScript is utilized to validate form inputs, enhancing user experience by providing immediate feedback and preventing erroneous data submissions. Additionally, XML can be used to structure and transport data, allowing for the design of web pages that efficiently handle and display complex information. By integrating these technologies, developers can create dynamic, well-structured, and user-friendly web applications.

Course Outcomes:
At the end of the course, the student will be able to
1. apply the fundamental HTML markups when designing web pages.
2. design the web pages with Tables and forms controls
3. implement cascading style sheets to design web pages
4. interpret and use JavaScript concepts in designing web pages.
5. apply XML features for developing web pages.

List of Programs

1) a) Develop an HTML page using various basic tags.
   b) Develop an HTML page using text processing tags.
2) Write an HTML page using Links concept.
3) Develop your class timetable using Tables concept.
4) Design Registration Form using various form controls.
5) Design a web page using Embedded Cascading Style Sheet.
6) Design a web page using External Cascading Style Sheet.
7) Develop a web page using various Lists.
8) Develop java script application to declare variables.
9) Design java script application using arrays concept
10) Write an XML file which will display the Books information and also XSLT

Program to display XML Data in HTML document which includes:
1) Title of the book 2) Author Name 3) Publisher 4) Edition 5) Price

Prescribed Textbooks:

1. Rob Larsen, *Beginning HTML and CSS* (Wrox Programmer to Programmer). Wiley, 2013.
2. Hans Bergstan. *Java Server Pages*. 3rd Ed., SPD publications, 2003.

Reference Books:

1. Jon Duckett, *Java Script and JQuery: Interactive Front-End Web Development*. Wiley Publications, 2014.
2. Jon Duckett, *Web Design with HTML CSS JavaScript and JQuery Set*. Wiley Publications, 2014.

Title of the Course: Design and Analysis of Algorithms Lab
Category: PC
Course Code: 24BCOM43L
Branches: **Computer Science**
Semester: IV Semester

Lecture Hours	Tutorial Hours	Practice Hours	Credits
0	0	4	2

Course Objectives: Students will be able to understanding, applying, and analyzing the performance of algorithmic techniques such as divide and conquer, greedy methods, dynamic programming, backtracking, and branch and bound to solve various computational problems. .

Course Outcomes:
At the end of the course, the student will be able to

1. understand the implementation of divide and conquer programming techniques .
2. apply the implementation greedy programming techniques
3. apply and Analyze the implementation dynamic programming techniques
4. analyze the implementation of back tracking techniques
5. analyze the implementation of branch and bound techniques

List of Programs

1. Exercise 1: Sort a given set of elements using the Quick sort for different values of n.
2. Exercise 2: Sort a given set of elements using the Merge sort for different values of n.
3. Exercise 3: Search for a particular element from a given set of elements using Binary Search technique.
4. Exercise 4: Find Minimum Cost Spanning Tree of a given undirected graph Prim's algorithm
5. Exercise 5: Find Minimum Cost Spanning Tree of a given undirected graph Kruskal's algorithm
6. Exercise 6: Implement the 0/1 knapsack problem by Dynamic programming algorithm
7. Exercise 7: Implement the Floyd Warshal's algorithm by dynamic programming technique.
8. Exercise 8: Consider the problem of four queens on an (4x4) chessboard. Two queens are said to attack each other if they are on the same row, column, or diagonal. Implements backtracking algorithm to solve the problem i.e. place Four non-attacking queens on the board.
9. Exercise 9: Find the Sum-of-Subset problem from the given set of elements by applying backtracking technique.
10. Exercise 10: Implement the graph coloring problem from the given number of vertices by applying backtracking technique.

Prescribed Textbooks:

1. Ellis Horowitz, Sartaj Sahni and S. Rajasekaram, *Fundamentals of Computer Algorithms*. 2nd Ed., Universities Press.

Reference Books:

1. R. C. T. Lee, S. S. Tseng, R. C. Chang and Y. T. Tsai, *Introduction to Design and analysis of Algorithms*: *A strategic approach*. McGraw Hill, 2005.
2. Aho, Ullman and Hopcroft, *Design and Analysis of algorithms. Pearson Education*, 2004.
3. M.T. Goodrich and R. Tomassia, *Algorithm Design: Foundations, Analysis and Internet Example*. John Wiley and sons.

Title of the Course: Introduction to Artificial Intelligence

Category: PC

Course Code: 24BAIL54T

Branches: **Computer Science, AI&ML, AI&DA, Programming & Data Science**

Semester: V Semester

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	0	0	3

Course Objectives: This course aims to provide a comprehensive understanding of the foundational building blocks of Artificial Intelligence (AI) through the concept of intelligent agents. It explores key approaches in AI, including heuristic search, game search, and logical inference, while delving into decision theory and planning processes for agents. Additionally, the course focuses on analyzing various types of objects and scenarios in uncertain environments and equipping learners with the ability to identify solutions using reasoning in the context of uncertain knowledge.

Course Outcomes:

At the end of the course, the student will be able to

1. Understand the importance of artificial Intelligence in real world environment
2. Apply the artificial intelligence algorithms for problem solving
3. Analyze the various reasoning and knowledge representation techniques
4. Solve the problems using classification and planning techniques
5. Apply knowledge and reasoning techniques in uncertain environment for the obtain solutions

Unit 1 Introduction to Artificial Intelligence 12

Introduction to AI, History of AI, Risks and Benefits of AI Emergence of Intelligent Agents, and Intelligent Agents: PEAS - Representation for an Agent, Types of Agents, Types of Agent Environments, Concept of Rational Agent, defining the Problem as a State Space Search, Problem Characteristics.

Unit 2 Problem Solving 12

Solving Problem by searching: Characteristics of problem solving, defining the Problem as a State Space Search, Problem solving Agents, Types of Search Techniques: Uninformed – BFS and DFS, Uniform Cost Search. Informed: Greedy Search, A* Algorithm, Hill Climbing.

Unit 3 Knowledge , Reasoning and Planning 12

Logical Agents: knowledge Based Agent, Architecture of Knowledge Based Agents, The wumpus World, Introduction to Logic, Propositional Logic. First Order Logic: Syntax and Semantics, Proportional vs. First Order Inference, Forward and Back ward chaining.

Unit 4 Knowledge Representation 12

Knowledge Representation: Ontological Engineering, Categories and Objects, Events, Mental Objects and Modal Logic. Automated Planning: Definition of classical planning, Algorithms for classical Planning, Heuristics for planning, Hierarchical planning.

Unit 5 Uncertain Knowledge and Reasoning 10

Quantifying Uncertainty: Acting Under Uncertainty, Basic Probability Notation, Probability axioms and their reasonableness, Inference Using Full Joint Distributions, Independence, Bayes Rule and Its Use.

Textbook:

1. Stuart Russell and Peter Norvig. Artificial Intelligence: A Modern Approach. 4th Ed., Pearson, 2020.

Reference Books:

1. George Lugar. AI-Structures and Strategies for Complex Problem Solving, 4th Ed., Pearson Education, 2002.
2. Patrick H. Winston. Artificial Intelligence. 3rd Ed., Pearson. 1992.

Title of the Course: Fundamentals of Big Data

Category: PC

Couse Code: 24BAIA53T

Branches **Computer Science, AI&ML, AI&DA, Programming & Data Science**

Semester: V

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	0	0	3

Course Objectives: This course introduces the basics of Big Data and Big Data analytics, including data types, features, challenges, and how it differs from traditional business intelligence. It explains Hadoop, HDFS, and MapReduce for storing and processing large data. The course also covers MapReduce programming and NoSQL databases. Students learn to use tools like Hive, MongoDB, Pig, and Cassandra for handling big data. Overall, the course helps students understand and use Big Data technologies in real-world applications.

Course Outcomes:

At the end of the course, the student will be able to

1. Explain digital data types, Big Data concepts, challenges, and the difference between traditional BI and Big Data analytics.
2. Describe Hadoop and HDFS architecture and explain how MapReduce processes data.
3. Use MapReduce programming concepts and explain the need and types of NoSQL databases.
4. Apply Cassandra and MongoDB to store, query, and analyze large-scale data.
5. Explain and use Hive and Pig for Big Data processing and distributed data storage.

Unit 1 **INTRODUCTION TO BIG DATA** 11

Types of Digital Data: Classification of Digital Data, Characteristics of Data, Evolution of Big Data, Definition of Big Data, Challenges with Big Data, What is Big Data?, Why Big Data?, Traditional Business Intelligence versus Big Data.
Big Data Analytics: What is Big Data Analytics, Why Big Data Analytics Important? Big Data Technologies, Classification of Analytics, Data Science, Terminologies used in Big Data Environments.

Unit 2 **BIG DATA TECHNOLOGIES** 12

Hadoop: History of Hadoop, Hadoop Distributions, Why Hadoop? Distribution Computing Challenges, Hadoop Distributed File System: HDFS Daemons, Anatomy of file Read, Anatomy of file Write, Processing Data with Hadoop: MapReduce Daemons, How does MapReduce work?, MapReduce Example

Unit 3 Introduction to Map Reduce Programming & NoSQL 12

Introduction to Map Reduce Programming: Introduction, Mapper, Reducer, Combiner, Partitioner.

NoSQL: What is NoSQL, Types of NoSQL Databases, Why NoSQL? Advantages of NoSQL, SQL versus NoSQL.

Unit 4 Introduction to MongoDB & Cassandra 12

MongoDB: What is MongoDB?, Why MongoDB?: Using JavaScript Object Notation(JSON), Creating or Generating a Unique Key, Support for Dynamic Queries, Storing Binary Data, Replication.

Cassandra: Apache Cassandra: An Introduction, Features of Cassandra: Peer-to-peer Network, Gossip and Failure Detection.

Unit 5 Introduction to Hive & PIG 11

Hive: What is Hive?, Hive Architecture, Hive Data Types, Hive File Format, Hive Query Language (HQL): Data Definition Language, Data Manipulation Language.

PIG: What is PIG? - Anatomy of Pig, Pig on Hadoop, Data types in Pig, Execution Modes of Pig.

Textbook:

1. Seema Acharya and Subhashini Chellappan. Big Data Analytics. 2nd Ed., Wiley, 2023.

Reference Books:

1. Michael Minelli, Michehe Chambers and Ambiga Dhiraj. Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today‟s Businesses, 1st Ed., Wiely CIO Series, 2013.
2. Arvind Sathi, Big Data Analytics: Disruptive Technologies for Changing the Game. 1st Ed., IBM Corporation, 2012.

Title of the Course: Advanced Python Programming
Category: PC
Couse Code: 24BCOM51T
Branches B.Sc. (Computer Science)
Semester: V Semester

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	0	0	3

Course Objectives: This course aims to provide Python programming, including variables, data types, operators, and input/output operations. It aims to develop students' problem-solving skills through the use of control statements and built-in data structures such as lists, tuples, strings, and dictionaries. The course also provides knowledge of arrays, NumPy, and file handling techniques for efficient data processing and data storage. Additionally, it enables students to design and organize programs using functions, modules, packages, and advanced functional programming concepts. Finally, the course familiarizes students with Object-Oriented Programming principles, exception handling, and regular expressions to support real-world application development.

Course Outcomes:

At the end of the course, the student will be able to

1. Apply fundamental Python concepts, including data types, operators, and input/output statements, to develop simple programs.
2. Use control structures and built-in data structures to efficiently solve computational problems.
3. Perform data manipulation and storage operations using arrays, NumPy, and file handling techniques.
4. Develop modular and reusable programs using functions, recursion, lambda expressions, generators, modules, and packages.
5. Design and implement object-oriented Python programs incorporating classes, inheritance, polymorphism, exception handling, and regular expressions.

Unit 1 Introduction to Python 12

Introduction python programming, variables and identifiers, Data Types, Operators in python and with examples, I/O functions, Operator precedence and Associativity rules with example programs.

Unit 2 Control Structures & List and String functions 13

Control statements: Decision control statements, looping statements and Jump control statements. List data type and its functions and Tuple data type and its conversion. Advantages of List and Tuple and

Comparison, String functions. Dictionaries with examples.

Unit 3 Arrays 12

What is Array, Advantages of Arrays, creating an Array, importing the Array module, indexing and Slicing on Arrays, Processing the Arrays, Types of Arrays, working with Arrays using numpy, mathematical operations on Arrays? Files in Python: Files, Types of files, working with files and working with binary files, with statement, The seek() and tell() methods.

Unit 4 Functions and modules 12

Defining functions, Returning Multiple Values From a function, Arguments in Functions, Variable scope, Recursive function, Anonymous Function, filer(), map(), reduce() functions. Function Decorators, Generators. Modules, Packages.

Unit 5 OOP's and Exception Handling 11

Introduction to Object Oriented Programming features of OOP's, Classes and Objects, Inheritance and Polymorphism. Exception Handling, Types of Exception Handling. Regular Expressions with examples.

Textbook:

1. Dr. R. Nageswar Rao. Core Python Programming. 3rd Ed., Dreamtech Press, 2021.

Reference Books:

1. Charles Dierbach. Introduction to Computer Science using Python. 2nd Ed., Wiley, 2012.

2. Wesley J Chun. Core Python Applications Programming. 3rd Ed., Prentice Hall, 2015.

ANNAMACHARYA UNIVERSITY

(ESTD UNDER AP PRIVATE UNIVERSITIES (ESTABLISHMENT AND REGULATION) ACT, 2016)
(UNIVERSITY LISTED IN UGC AS PER THE SECTION 2(f) OF THE UGC ACT, 1956)

RAJAMPET, Annamayya District, AP – 516126, INDIA

Title of the Course: Differential Equations

Category: BS

Course Code: 24BMAT51T

Branches: **Computer Science, AI&ML, AI&DA, Programming & Data Science**

Semester: V Semester

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	0	0	3

Course Objectives: This course aims to develop fundamental understanding of ordinary and partial differential equations and their solution techniques. It enables students to handle first-order and higher-order differential equations and apply them to physical and engineering problems. The course also introduces methods for solving partial differential equations arising in practical situations.

Course Outcomes:

At the end of the course, the student will be able to

1. Solve first-order differential equations using standard solution methods.
2. Determine complementary functions and particular integrals of higher-order linear differential equations.
3. Compute particular integrals using operator methods and variation of parameters.
4. Apply differential equations to model growth, cooling, motion, and electrical circuits.
5. Solve first-order partial differential equations using standard techniques

Unit 1 First Order Differential Equations 11

Formation of differential equations, First order and first-degree equations, Solution of a differential equations-Variables separable, Homogeneous equations, Exact differential equations, Linear differential equations, Bernoulli's equation.

Unit 2 Applications of first order Differential Equations 12

Applications of first order Differential Equations- Orthogonal trajectories (in cartesian form), Newton's law of cooling, Law of natural growth or decay

Unit 3 Linear differential equations of higher order with constant coefficients 12

Basic concepts - general solution-operator D-rules for finding complimentary function-inverse operator-rules for finding particular integral for RHS term of the type e^{ax}, $\sin ax / \cos ax$, polynomials in x, $e^{ax} sin\, a\, x / e^{ax} cos\, a\, x / e^{ax} x^n$, Method of variation of parameters

Unit 4 Linear differential equations of higher order with variable coefficients 11

Linear differential equations with variable coefficients, The Cauchy - Euler Equation, Legendre's Linear Equations. Applications: Electric Circuits-L-C and L-C-R circuit problems.

Unit 5 Partial Differential Equations 12

Formation of Partial Differential Equations PDEs, Linear PDEs of first order, Lagrange's method, Charpit's method (simple cases).

Textbook:

1. B.S. Grewal. *Higher Engineering Mathematics.* 44th Ed., Khanna Publishers, 2017.

Reference Books:

1. G. B. Thomas, Maurice D. Weir and Joel Hass. *Thomas' Calculus*, 14th Ed., Pearson Publishers, 2022.
2. R.K. Jain, S.R.K. Iyengar. *Advanced Engineering Mathematics.* 5th Ed., Narosa Publications.

Title of the Course: Software Architecture

Category: PE

Couse Code: 24BCOM5AT

Branch B.Sc.(Computer Science)

Semester: V Semester

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	0	0	3

Course Objectives: This course introduces the fundamentals of software architecture, its styles, and design principles. It focuses on the role of architecture in software processes, business goals, and system quality. Students learn shared information systems, user-interface architecture guidance, and architectural description languages. The course also covers architectural tools, styles, and environments for effective design and analysis.

Course Outcomes: At the end of the course, the student will be able to

1. Explain the importance of architecture in software processes
2. Analyze various architectural styles such as pipes and filters
3. Analyze the different database integration approaches and architectural structures
4. Apply the guidance for user-interface architectures and linguistic requirements
5. Analyze architectural design tools and environments, focusing on UniCon.

Unit 1 **INTRODUCTION TO SOFTWARE ARCHITECTURE** 12

INTRODUCTION TO SOFTWARE ARCHITECTURE: what is software Architecture? An Engineering Discipline for Software, Status of Software Architecture, Architecture Business Cycle, Importance of Architectures, Software Processes and the Architecture Business Cycle, Features of Good Architecture.

Unit 2 **ARCHITECTURE STYLES** 11

ARCHITECTURE STYLES: Pipes and Filters, Data Abstraction and Object Oriented organization, Even-based Implicit Invocation, Layered Systems, Repositories, Interpreters, Process Control, Heterogeneous Architectures.

Unit 3 **SHARED INFORMATION SYSTEMS** 12

SHARED INFORMATION SYSTEMS: Database Integration: Batch Sequential, Simple Repository, Virtual Repository, Hierarchical Layers. Integration in Software Development Environments, Architectural Structures for Shared Information Systems: Various on Dataflow Systems, variants on Repositories.

Unit 4 **GUIDANCE FOR USER-INTERFACE ARCHITECTURE & LINGUISTIC ISSUES** 12

Guidance for user-Interface Architecture: Design Space Rules, Design Space for User-Interface Architectures, Design Rules for User-Interface Architectures. Linguistic Issues: Requirements for Architecture – description Languages: Linguistic Character of Architectural Description, Desiderata for Architectural-Description Languages, And Problem with Existing Languages

Unit 5 **TOOLS FOR ARCHITECTURAL DESIGN** 12

UniCon: A universal Connector Language, Components and connects, Abstraction and encapsulation, Types and Type Checking, Accommodating Analysis tool. Architectural Design Environment: What is Architectural Style? Automated Support for Architectural Design, Observations about Environments for Architectural Design.

Textbook:

1. Mary Shaw and David Garlan. SOFTWARE ARCHITECTURE- Perspective on an Emerging Discipline. 1st Impression ED., Pearson, 2015.

Reference Books:

1. Len Bass, Paul Elements and Rick Kazman. Software Architecture in Practice. PEARSON, 4th Ed., 2021.
2. Buschmann. Pattern Oriented Software Architecture. Wiley, 1996.

Title of the Course: Quantum Computing
Category: PE
Couse Code: 24BCOM5BT
Branch: **Computer Science, AI&ML, AI&DA, Programming & Data Science**
Semester: V Semester

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	0	0	3

Course Objectives:

This course aims to introduce the interdisciplinary foundations of quantum computing, develop mathematical and physical understanding of qubits and quantum circuits, explore major quantum algorithms and complexity concepts, and provide knowledge of quantum error correction, information theory, and cryptographic applications.

Understand the fundamentals of linear algebra, Hilbert space, quantum physics principles

Course Outcomes

1. Explain the history and interdisciplinary importance of quantum computing
2. Apply mathematical and physical concepts relevant to quantum systems.
3. Understand physical implementation of Qubit
4. Analyze quantum algorithms and their computational advantages.
5. Understand the Impact of Quantum Computing on Cryptography

UNIT 1 **11**

History of Quantum Computing: Importance of Mathematics, Physics and Biology. Introduction to Quantum Computing: Bits Vs Qubits, Classical Vs Quantum logical operations

UNIT 2 12

Background Mathematics: Basics of Linear Algebra, Hilbert space, Probabilities and measurements. Background Physics: Paul's exclusion Principle, Superposition, Entanglement and super-symmetry, density operators and correlation, basics of quantum mechanics, Measurements in bases other than computational basis. Background Biology: Basic concepts of Genomics and Proteomics (Central Dogma)

UNIT 3 11

Qubit: Physical implementations of Qubit. Qubit as a quantum unit of information. The Bloch sphere Quantum Circuits: single qubit gates, multiple qubit gates, designing the quantum circuits. Bell states.

UNIT 4 11

Quantum Algorithms: Classical computation on quantum computers. Relationship between quantum and classical complexity classes. Deutsch's algorithm, Deutsch's-Jozsa algorithm, Shor's factorization algorithm, Grover's search algorithm.

UNIT 5 12

Noise and error correction: Graph states and codes, Quantum error correction, fault-tolerant computation. Quantum Information and Cryptography: Comparison between classical and quantum information theory. Quantum Cryptography, Quantum teleportation.

Text Book:

1. Nielsen M. A., Quantum Computation and Quantum Information, Cambridge.

Reference Books:

1. Quantum Computing for Computer Scientists by Noson S. Yanofsky and Mirco A. Mannucci

2. Benenti G., Casati G. and Strini G., Principles of Quantum Computation and Information, Vol.I: Basic Concepts, Vol II.

Title of the Course: Natural Language Processing
Category: PE
Couse Code: 24BCOM5CT
Branch B.Sc. (Computer Science
Semester: V Semester

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	-	-	3

Course Objectives: The learner will be able to explain and apply fundamental algorithms and techniques in Natural Language Processing (NLP), demonstrating a clear understanding of core concepts and practical implementations. They will develop knowledge of syntactic and semantic approaches used in NLP for analyzing language structure and meaning. The learner will also understand statistical methods employed in machine translation and gain insight into modern language modeling techniques used for processing and generating human language effectively.

Course Outcomes: At the end of the course, the student will be able to

1. Explain core NLP concepts and linguistic foundations.
2. Apply grammar and parsing techniques for syntactic analysis.
3. Analyze language structures using advanced parsing methods.
4. Interpret semantic representations and resolve ambiguity.
5. Develop and evaluate language models for multilingual applications.

Unit 1 Introduction to Natural language 12

The Study of Language, Applications of NLP, Evaluating Language Understanding Systems, Different Levels of Language Analysis, Representations and Understanding, Organization of Natural language Understanding Systems, Linguistic Background: An outline of English Syntax.

Unit 2 Grammars and Parsing 11

Grammars and Parsing- Top- Down and Bottom-Up Parsers, Transition Network Grammars, Feature Systems and Augmented Grammars, Morphological Analysis and the Lexicon, Parsing with Features, Augmented Transition Networks.

Unit 3 Grammars for Natural Language 12

Grammars for Natural Language, Movement Phenomenon in Language, Handling questions in Context Free Grammars, Human Preferences in Parsing, Shift Reduce Parsers, Deterministic Parsers.

Unit 4 Semantic Interpretation 12

Semantic Interpretation: Semantic & Logical form, Word senses & ambiguity, The basic logical form language, Encoding ambiguity in the logical Form, Verbs & States in logical form, Thematic roles, Speech acts & embedded sentences, Defining semantics structure model theory.

Unit 5 Linking Syntax and Semantics 12

Linking Syntax and Semantics: semantic interpretation and Compositionality, A simple Grammar and Lexicon with semantic Interpretation, Prepositional Phrase and Verb Phrases, Lexicalized semantic Interpretation and Semantic Roles.

Textbook:

1. James Allen. Natural Language Understanding. 2nd Ed., Pearson, 2003.

Reference Books:

1. Daniel M.Bikel and ImedZitouni. Multilingual Natural Language Processing Applications from Theory to Practice. 1st Ed., Pearson, 2012.
2. Jurafsky, Dan and Martin, James. Speech and Language Processing. 2nd Ed., Prentice Hall, 2008.

Title of the Course: Artificial Intelligence Lab

Category: PC

Course Code: 24BAIL54L

Branches: B.Sc. (Computer Science)

Semester: V Semester

Lecture Hours	Tutorial Hours	Practice Hours	Credits
0	0	4	2

Course Objectives: This course is designed to equip students with the knowledge and skills to analyze various artificial intelligence (AI) techniques and methodologies. Students will gain a comprehensive understanding of implementing key AI approaches, including heuristic search methods and techniques for solving constraint satisfaction problems. Additionally, the course focuses on the implementation of game-playing algorithms, enabling students to design intelligent systems capable of strategic decision- making. Through practical applications and problem-solving, students will develop a strong foundation in core AI concepts and their real-world applications.

Course Outcomes:

At the end of the course, the student will be able to

1. Understand and implement fundamental AI concepts, intelligent agents, and basic problem-solving techniques.
2. Apply uninformed search algorithms like Depth First Search (DFS) and Breadth First Search (BFS) to solve problems.
3. Analyze and implement informed search techniques such as A* and Hill Climbing algorithms.
4. Develop solutions for classical AI problems like Water Jug, 8-Puzzle, and Crypt Arithmetic using appropriate methods.
5. Apply probability concepts, independence, and Bayes Rule to handle uncertainty in decision-making

List of Experiments:

1. Write a program to implement chat box.
2. Write a program to solve any problem using Depth First Search.
3. Write a program to solve any problem using Breadth First Search.
4. Write a program to implement Water jug problem.
5. Write a program to implement A* search.
6. Write a program to implement Crypt arithmetic using Constraint satisfaction problem.
8. Write a program to implement 8-Puzzle game.
9. Write a program for probability and Independence check.
10. Write a program for Bayes Rule.

Textbook:

1. Stuart Russell and Peter Norvig. Artificial Intelligence: A Modern Approach. 2nd Ed., Pearson Publication, 2020.

Title of the Course: Advanced Python Programming Lab

Category: PC

Couse Code: 24BCOM51L

Branch B.Sc.(Computer Science) Programming & Data Science

Semester: V Semester

Lecture Hours	Tutorial Hours	Practice Hours	Credits
0	0	4	2

Course Objectives: This course aims to provide Python programming, including variables, data types, operators, and input/output operations. It aims to develop students' problem-solving skills through the use of control statements and built-in data structures such as lists, tuples, strings, and dictionaries. The course also provides knowledge of arrays, NumPy, and file handling techniques for efficient data processing and data storage. Additionally, it enables students to design and organize programs using functions, modules, packages, and advanced functional programming concepts. Finally, the course familiarizes students with Object-Oriented Programming principles, exception handling, and regular expressions to support real-world application development.

Course Outcomes:

At the end of the course, the student will be able to

1. Apply fundamental Python concepts, including data types, operators, and input/output statements, to develop simple programs.
2. Use control structures and built-in data structures to efficiently solve computational problems.
3. Perform data manipulation and storage operations using arrays, NumPy, and file handling techniques.
4. Develop modular and reusable programs using functions, recursion, lambda expressions, generators, modules, and packages.
5. Design and implement object-oriented Python programs incorporating classes, inheritance, polymorphism, exception handling, and regular expressions.

List of Experiments:

Note: Any 10 experiments to be performed

1. Write a python program to implement various operator and precedence rules?
2. Write a python program to implement various control statement
3. Write a python program to implement various looping statements

4. Write a python program to implement various functions in list and tuple

5. Write a python program to implement various string handling functions

6. Write a python program to implement various methods in dictionary data type

7. a) Write a python program to understand various methods of arrays class

b) Write a python program to storing student's marks into an array and finding total and total marks and percentage

c) Write a python program to accept two matrices and find their product?

8. Write a Python programs to implement various functions and modules and using recursive functions

9. Write a Python programs to implement various file operations

10. Write a python program to implement function Decorators and generators and using modules

11. Write a python program to classes and objects and using oop's concepts

12. Write a python program to implement various exception handling methods and Regular expressions?

Textbook:

1. Dr. R. Nageswar Rao. Core Python Programming. 3rd Ed., Dreamtech Press, 2021.

Title of the Course: Soft Skills
Category: SEC
Couse Code: 24BENG51S
Branch/es: **Computer Science, AI&ML, AI&DA, Programming & Data Science**
Semester: V Semester

Lecture Hours	Tutorial Hours	Practice Hours	Credits
1	0	2	2

Course Objectives:
This course aims to promote all-round development of students by strengthening soft skills, developing critical thinking and problem-solving abilities, fostering healthy interpersonal relationships, and enabling effective collaboration within diverse and heterogeneous teams in professional environments.

Course Outcomes:
At the end of the course, the student will be able to

1. Describe methods for building professional image
2. Apply critical thinking skills in problem solving
3. Analyze the needs of an individual and team for well-being
4. Evaluate the situation and take necessary decisions
5. Create a productive workplace atmosphere using social and work-life skills ensuring personal and emotional well-being

Unit 1 12

Soft Skills & Communication Skills

Soft Skills - Introduction, Need - Mastering Techniques of Soft Skills – Communication Skills -Significance, Process, Types - Barriers to Communication - Improvising Techniques.

Activities:

Intrapersonal Skills- Narration about self- strengths and weaknesses- Clarity of thought – Self- expression – Articulating with felicity.
(The facilitator can guide the participants before the activity citing examples from the lives of the great, anecdotes and literary sources)
Interpersonal Skills- Group Discussion – Debate – Team Tasks - Book and film Reviews by groups - Group leader presenting views (non- controversial and secular) on contemporary issues or on a given topic.
Verbal Communication- Oral Presentations- Extempore- Brief addresses and speeches- Convincing- Negotiating- Agreeing and disagreeing with professional grace.
Non-verbal communication – Public speaking – Mock interviews – Presentations with an objective to identify non- verbal clues and remedy the lapses on observation.

Unit 2 11

Critical Thinking

Active Listening – Observation – Curiosity – Introspection – Analytical Thinking – Open-mindedness – Creative Thinking - Positive Thinking - Reflection

Activities:

Gathering information and statistics on a topic - Sequencing – Asserting – Reasoning – Critiquing issues – Placing the problem – Finding the root cause - Seeking viable solution – Judging with rationale – Evaluating the views of others - Case Study, Story Analysis

Unit 3 12

Problem Solving & Decision Making

Meaning & Features of Problem Solving – Managing Conflict – Conflict resolution – Team building - Effective decision making in teams – Methods & Styles

Activities:

Placing a problem which involves conflict of interests, choice and views – Formulating the problem – Exploring solutions by proper reasoning – Discussion on important professional, career and organizational decisions and initiate debate on the appropriateness of the decision.

Case Study & Group Discussion

Unit 4 11

Emotional Intelligence & Stress Management

Managing Emotions – Thinking before Reacting – Empathy for Others – Self-awareness – Self-Regulation – Stress factors – Controlling Stress – Tips

Activities:

Providing situations for the participants to express emotions such as happiness, enthusiasm, gratitude, sympathy, and confidence, compassion in the form of written or oral presentations.

Providing opportunities for the participants to narrate certain crisis and stress –ridden situations caused by failure, anger, jealousy, resentment and frustration in the form of written and oral presentation, organizing debates

Unit 5 11

Corporate Etiquette

Etiquette- Introduction, Concept, Significance - Corporate Etiquette - meaning, modern etiquette, benefits - Global and Local Culture Sensitivity - Gender Sensitivity - Etiquette in Interaction- Cell phone Etiquette - Dining Etiquette - Netiquette - Job Interview Etiquette - Corporate Grooming tips -Overcoming challenges

Activities

Providing situations to take part in the Role Plays where the students will learn about bad and good manners and etiquette - Group Activities to showcase gender sensitivity, dining etiquette etc. - Conducting mock job interviews - Case Study - Business Etiquette Games

NOTE-:

1. The facilitator can guide the participants before the activity citing examples from the lives of the great, anecdotes, epics, scriptures, autobiographies and literary sources which bear true

relevance to the prescribed skill.

2. Case studies may be given wherever feasible for example for Decision Making- The decision of King Lear.

Textbooks:

1. Barun K. Mitra. *Personality Development and Soft Skills*. Paperback Ed., Oxford University Press, 2012.
2. Shikha Kapoor. *Personality Development and Soft Skills: Preparing for Tomorrow*. I K International Publishing House, 2018.

Reference Books:

1. Prashant Sharma. *Soft Skills: Personality Development for Life Success*. BPB Publications, 2018.
2. Alex K. *Soft Skills*. Revised Ed., S. Chand & Co., 2012.
3. Gajendra Singh Chauhan and Sangeetha Sharma. *Soft Skills: An Integrated Approach to Maximize Personality*. Wiley, 2013.
4. Sabina Pillai and Agna Fernandez. *Soft Skills and Employability Skills*. Cambridge University Press, 2018.
5. Renu Shorey. *Soft Skills for a Big Impact*. Notion Press.
6. Rajiv Kumar Jain and Usha Jain. *Life Skills*. Vayu Education of India, 2014.

Web Resources:

GRAMMAR:

1. https://youtu.be/DUlsNJtg2L8?list=PLLy_2iUCG87CQhELCytvXh0E_y-bOO1_q
2. https://youtu.be/xBaLgJZ0t6A?list=PLzf4HHlsQFwJZel_j2PUy0pwjVUgj7KlJ
3. https://youtu.be/-Y-R9hDl7lU
4. https://youtu.be/gkLsn4ddmTs
5. https://youtu.be/2bf9K2rRWwo
6. https://youtu.be/FchfE3c2jzc
7. https://www.businesstrainingworks.com/training-resource/five-free-business-etiquette-training-games/
8. https://onlinecourses.nptel.ac.in/noc24_hs15/preview
9. https://onlinecourses.nptel.ac.in/noc21_hs76/preview

Title of the Course: Distributed Systems

Category: PE

Couse Code: 24BCOM6AT

Branch B.Sc.(Computer Science)

Semester: VI Semester

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	0	0	3

Course Objectives: This course aims to provide a comprehensive understanding of distributed systems, including their characteristics, system models, and associated challenges. It enables students to analyze concepts such as time synchronization, global states, coordination, and agreement in distributed environments. The course develops knowledge of inter-process communication, distributed objects, remote invocation, and distributed file and naming services. It also focuses on distributed shared memory and consistency models. Finally, it equips students with the principles of transactions, concurrency control, and distributed transaction management to design reliable and scalable distributed systems.

Course Outcomes:

At the end of the course, the student will be able to

1. Explain the fundamental concepts and characteristics of distributed systems, including real-world examples and applications.
2. Explain the concepts of time, events, process states, and global states in distributed systems.
3. Apply techniques for external data representation and marshalling to enable communication between heterogeneous systems.
4. Apply knowledge of distributed file systems, name services, and DSM to design scalable, consistent, and efficient distributed systems.
5. Apply transaction management and concurrency control strategies to design reliable, consistent, and fault-tolerant distributed systems.

UNIT 1 CHARACTERIZATION OF DISTRIBUTED SYSTEMS 12

Introduction, Examples of Distributed Systems, Resource Sharing and the Web, Challenges.
System Models: Introduction, Architectural Models, Fundamental Models.

UNIT 2 TIME AND GLOBAL STATES 12

Introduction, Clocks Events and Process States, Synchronizing Physical Clocks, Logical Time and Logical Clocks, Global States, Distributed Debugging.
Coordination and Agreement: Introduction, Distributed Mutual Exclusion, Elections, Multicast Communication, Consensus and Related Problems.

UNIT 3 INTER PROCESS COMMUNICATION 12

Introduction, The API for the Internet Protocols, External Data Representation and Marshalling, Client-Server Communication, Group Communication.
Distributed Objects and Remote Invocation: Introduction, Communication between Distributed Objects, Remote Procedure Call, Events and Notifications.

UNIT 4 DISTRIBUTED FILE SYSTEMS 11

Introduction, File Service Architecture, Name Services: Introduction, Name Services and the Domain Name System, Directory Services, Distributed Shared Memory: Introduction, Design and Implementation Issues, Release Consistency, Other Consistency Models.

UNIT 5 TRANSACTIONS AND CONCURRENCY CONTROL 11

Introduction, Transactions, Nested Transactions, Locks, Optimistic Concurrency Control, Timestamp Ordering, Comparison of Methods for Concurrency Control. Distributed Transactions: Introduction, Flat and Nested Distributed Transactions, Atomic Commit, Protocols, Concurrency Control in Distributed Transactions, Distributed Deadlocks, Transaction Recovery.

Textbook:

1. George Coulouris, J Dollimore and Tim Kindberg. Distributed Systems- Concepts and Design. 4th Ed., Pearson Education, 2011.

Reference Books:

1. Andrew S. Tanenbaum and Maarten Van Steen. Distributed Systems- Principles and Paradigms. 2nd Ed., Pearson Education**,** 2007.

2. Sukumar Ghosh. Distributed Systems-An Algorithm Approach. 2nd Ed., Chapman & Hall/CRC, Taylor & Fransis Group, 2015.

Title of the Course: Web Services

Category: PE

Couse Code: 24BCOM6BT

Branches B.Sc.(Computer Science)

Semester: VI Semester

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	0	0	3

Course Objectives: This course aims to provide students with a comprehensive understanding of distributed computing and the evolution of web services, including their definition, architecture, standards, and challenges. It equips learners with knowledge of key distributed computing technologies such as Client/Server, CORBA, Java RMI, DCOM, and MOM. The course emphasizes the design, development, and implementation of web services using SOAP, WSDL, and UDDI, covering communication models, security, and service discovery. Students will gain practical skills in building web service–enabled applications and working with service registries. Overall, the course prepares learners to develop scalable, interoperable, and standards-compliant web services.

Course Outcomes:

At the end of the course, the student will be able to

1. Explain the concept of distributed computing and its significance in modern computing environments.
2. Explain the architecture of web services and its core building blocks, including service providers, and registries.
3. Apply SOAP security mechanisms to ensure secure web service interactions.
4. Analyze the applicability, limitations, and future trends of WSDL in web service development.
5. Analyze the limitations and challenges of UDDI while designing and managing discoverable web services.

UNIT 1 12

EVOLUTION AND EMERGENCE OF WEB SERVICES – What is distributed computing? Importance of distributed computing technologies – client/server, CORBA, JAVA RMI, Micro Soft DCOM, MOM, Challenges in Distributed Computing, Introduction to Web Services – The definition of web services, basic operational model of web services, Core web services Standards, Other Industry Standards Supporting Web Services, Challenges in using web services.

UNIT 2 WEB SERVICES ARCHITECTURE AND TECHNOLOGIES 11

Web Services Architecture and its Core Building Blocks, Tools of Trade, Web Services Communication Models, Implementing Web Services, Developing Web Services Enabled Applications.

UNIT 3 DEVELOPING WEB SERVICES USING SOAP 12

XML Based Protocols and SOAP, Anatomy of SOAP Message, SOAP Encoding, SOAP Message Exchange Model, SOAP Communication, SOAP Messaging, SOAP Bindings for Transport Protocols, SOAP Security, Building SOAP Web Services.

UNIT 4 DESCRIPTION OF WEB SERVICES 11

Web Services Description Language(WSDL), WSDL in the World of Web Services, Anatomy of a WSDL Definition Document, WSDL Bindings, WSDL Tools, Future of WSDL, Limitations of WSDL.

UNIT 5 DISCOVERY OF WEB SERVICES 12

Universal Description, Discovery and Integration(UDDI), UDDI Registries, Programming with UDDI, Inquiry API, Publishing API, Implementations of UDDI, Registering as a Systinet UDDI Registry User, Publishing Information to a UDDI Registry, Searching Information in a UDDI Registry, Deleting Information from a UDDI Registry, Limitations of UDDI.

Textbook:

1. R. Nagappan, R. Skoczylas and R.P. Sriganesh. Developing Java Web Services. 1st Ed., Wiley India, 2008.

Reference Books:

1. F.P. Coyle. XML- Web Services, and the Data Revolution. 1st Ed., Pearson Education, 2002.
2. S. Graham and others. Building web Services with Java, 2nd Edition., Pearson Education, 2008.
3. D.A. Chappell & T. Jewell. Java Web Services. 1st Ed., O" Reilly, 2002.
4. McGovern, et al. Java web Services Architecture. 1st Ed., Morgan Kaufmann Publishers, 2003.

Title of the Course: Information Retrieval Systems

Category: PE

Couse Code: 24BCOM6CT

Branches B.Sc. (Computer Science), AI&DA

Semester: VI Semester

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	0	0	3

Course Objectives: The objective of this course is to cover the fundamentals of Information retrieval, including its history, key concepts, user tasks, and system architecture. IR models such as Boolean, vector, and probabilistic models, and their applications. The metrics are like recall and precision, query languages, reference collections. Document pre-processing, metadata, markup languages, multimedia retrieval, and indexing techniques. The parallel and distributed IR systems, online retrieval platforms, natural language search and digital library architectures

Course Outcomes:

At the end of the course, the student will be able to

1. Understand the fundamentals of IR, history, key concepts, tasks and system architecture.
2. Explain different IR models such as Boolean, vector, and probabilistic models, and their applications.
3. Apply evaluation metrics like recall and precision and **use** query languages and reference collections.
4. Explain document preprocessing, metadata, markup languages, multimedia retrieval techniques.
5. Explain parallel and distributed IR, online retrievals, natural language search and architectures.

Unit 1 INTRODUCTION 12

Information Retrieval: Definition, Information versus Data Retrieval, Early Developments, The IR Problem, The Users Task, The Software Architecture of the IR System.
Modeling: Taxonomy of IR Models, Ad hoc and Filtering, Characterization of IR Model, Classic IR Retrieval Basics(Boolean, Vector, Probabilistic Models Basics)

Unit 2 RETRIEVAL EVALUATION AND QUERY LANGUAGE 12

Retrieval Evaluation: Recall and Precision, Reference Collections: TREC,CACM,ISI and Cystic Fibrosis collections **Query Language:** Keyword-Based Querying, Pattern Matching, Structural Queries, Query Protocols, Trends and Research Issues

Unit 3 TEXT, MULTIMEDIA LANGUAGE AND MULTIMEDIA IR 11

Text: Metadata, Markup Languages, Multimedia, Document Pre-processing: Lexical, Elimination of stopwords, Stemming, Index Terms and Thesauri

Multimedia IR: Data Modeling, Query Languages: Request Specification, Conditions on Multimedia Data, Query Expressions

Unit 4 PARALLEL, DISTRIBUTED IR and The Web 12

Parallel Computing, Performance Measures, Parallel IR: MIMD and SIMD Architectures, Distributed IR: Introduction, Collection Partitioning, Source Selection, Query Processing, The Web: Challenges, Characterizing the Web, Browsing, Web Issues

Unit 5 IR BIBLIOGRAPHICAL SYSTEMS AND DIGITAL LIBRARIES 12

Online IR Systems and Document Databases: Databases, Online Retrieval Systems, IR in online Retrieval Systems, Natural Language Searching, Online Public Access Catalogs(OPACs)

Digital Libraries: Definition, Architectural Issues, Document Models, Representation and Access

Textbook:

1. Ricardo Baeza-Yates and Berthier RiBiero-Neto. Modern Information Retrieval. 19th Impression., Pearson, 2023.

Reference Books:

1. David A.Grossman and Ophir Frieder. Information Retrieval. 2nd Ed., Springer, 2008.

Title of the Course: Transform Techniques

Category: BS

Course Code: 24BMAT61T

Branches: **Computer Science, AI&ML, AI&DA, Programming & Data Science**

Semester: VI Semester

Lecture Hours	Tutorial Hours	Practice Hours	Credits
3	0	0	3

Course Objectives: This course introduces Laplace and Z–transforms and their properties for solving differential and difference equations. It develops skills in inverse transforms and emphasizes applications in engineering and scientific problems. The course enables students to apply transform techniques to simplify and solve initial value problems and discrete systems.

Course Outcomes:

At the end of the course, the student will be able to

1. Compute Laplace transforms using standard properties.
2. *Determine inverse Laplace transforms using standard methods*.
3. *Apply Laplace transforms to solve differential equations*.
4. Evaluate Z–transforms of standard sequences.
5. Solve difference equations using inverse Z–transforms.

Unit 1 Laplace Transform 12

Definition of Laplace Transform, Existence of Laplace Transform, Laplace Transform of standard functions, Linearity property, First shifting theorem, multiplication by t^n, division by t.

Unit 2 Inverse Laplace Transform 12

Definition of Inverse Laplace Transform, Inverse Laplace Transform of standard forms, Inverse Laplace Transform using partial fractions, Convolution theorem (without proof)..

Unit 3 Applications of Laplace Transform 11

Laplace Transform of derivatives, Laplace Transform of integrals, Solution of simple ordinary differential equations, Initial value theorem

Unit 4 Z–Transform 12

Definition of Z–Transform, Z–Transform of standard sequences, Linearity property, Shifting property.

Unit 5 Inverse Z–Transform and Applications 12

Inverse Z–Transform (partial fraction method), Initial value theorem, Solution of simple difference equations.

Textbook:

1. B.S. Grewal. *Higher Engineering Mathematics.* 44th Ed., Khanna Publishers, 2017.

Reference Books:

1. G. B. Thomas, Maurice D. Weir and Joel Hass. *Thomas' Calculus*, 14th Ed., Pearson Publishers, 2022.
2. R.K. Jain, S.R.K. Iyengar. *Advanced Engineering Mathematics.* 5th Ed., Narosa Publications.