SNDT WOMEN'S UNIVERSITY
Bachelor of Computer Applications (BCA)

AICTE RECOGNIZED
SYLLABUS FOR THE ACADEMIC YEAR 2024-2025

Terminologies

Abbreviation	Full-form	Remarks	Related to Major and Minor Courses
Major (Core)	MainDiscipline		
Major (Elective)	Elective Options		related to the Major Discipline
Minor Stream	Other Disciplines (Inter/ Multidisciplinary) nonrelated the Major	either from the same Faculty or any other faculty	
OEC	Open Elective Courses/ Generic		Not Related to The Major and Minor
VSEC	Vocational and Skill Enhancement Courses		
VSC	Vocational Skill Courses		Not Related to The Major and Minor
SEC	Skill Enhancement Courses		Not Related to The Major and Minor
AEC	Ability Enhancement Courses	Communication skills, critical reading, academic writing, etc.	Not Related To the Major and Minor

VEC	Value Education Courses	Understanding India, Environmental science /Education, Digital and technological solutions, Health & Wellness, Yoga education, sports, and fitness	Not Related To the Major and Minor

IKS	Indian Knowledge System	I. Generic IKS Course: basic knowledge of the IKS II. Subject Specific IKS Courses: advanced information pertaining to the subject: part of the major credit.	Subject Specific IKS related to Major
OJT	On-Job Training(Internship/Apprenticeship)	Corresponding to the Major subject	Related to the Major
FP	Field projects	corresponding to the Major Subject	Related to the Major
CC	Co-curricular Courses	Health and Wellness, Yoga education sports, and fitness, Cultural Activities, NSS/NCC and Fine/Applied/Visual/Performing Arts	Not Related To the Major and Minor
CE	Community Engagement and service		Not Related to the Major and Minor
RP	Research Project	Corresponding to the Major Subject	Related to the Major

Programme Template:

Programme Degree B.A./B.Com./B.Sc./B.M.S., etc.		**Bachelor of Computer Application (BCA)**
Parenthesis if any (Specialization) e.g. History, Human Development, English, etc.		
Preamble (Brief Introduction to the programme)		The Bachelor of Computer Applications (BCA) program is a four-year undergraduate degree program as per NEP-2020 designed to provide students with a strong foundation in computer science and its applications. The program aims to equip students with the knowledge and skills required to excel in the rapidly evolving field of computer science and information technology. The BCA program combines theoretical knowledge with practical applications to ensure that students develop a comprehensive understanding of computer systems, software development, database management, networking, and other core areas of computer science. It is an ideal choice for students who are interested in pursuing a career in the IT industry or furthering their studies in computer science. During the course of the BCA program, students are exposed to a wide range of subjects that cover various aspects of computer science. These subjects typically include programming languages, data structures, algorithms, computer architecture, operating systems, software engineering, web development, database management systems, computer networks, and information security. Upon successful completion of the BCA program, graduates have a wide range of career opportunities in the IT industry. They can work as software developers, system analysts, database administrators, network administrators, web developers, IT consultants, and other related roles. Graduates may also

		choose to pursue higher education, such as a Master's degree in computer science or a specialized field within the IT domain. By combining theoretical knowledge, practical skills, and industry exposure, the program equips students with the necessary tools to thrive in the IT industry and contribute to technological advancements.
Programme Outcomes (POs) *Action Verbs demonstrating*		After completing this programme, Learner will
	1.	Describe a strong foundation in computer application, including knowledge of Programming languages, Database,

(Major) discipline-related knowledge acquisition, mastery over cognitive and professional, vocational skills are to be used e.g. demonstrate sound understanding of, analyses, compare, create, design, etc…(minimum5)		Mathematics, Operating system and Networking.
	2.	Analyze the ethical and professional responsibilities in the field of computer applications by evaluating the implications of adhering to professional standards and practices.
	3.	Applying programming knowledge to develop a software application to solve specific problems.
	4.	Evaluate software designs and architectures for efficiency, security and user experience.
	5.	Design a software application to meet the requirements of the Industrial Standards.
Eligibility Criteria for Programme		As per DTE norms & State CET Cell
Intake (For SNDTWU Departments and Conducted Colleges)		As per AICTE approval norms

Structure with Course Titles
(Options related to our area of study to be provided with "OR" for baskets of different types)

SN	Courses	Type of Course	L	T	P	Credits	Marks	Int	Ext
	Semester I								
10035411	Problem Solving Techniques	Major(Core)	2	0	4	4	100	50	50
10035412	Mathematics Foundations to Computer Science - I	Major(Core)	2	0	0	2	50	0	50
10435411	Digital Marketing	OEC	2	0	4	4	100	50	50
10635401	Computer Architecture	VSC	1	0	2	2	50	50	0
10735401	Web Technology - I	SEC	1	0	2	2	50	50	0
Univ Code	General English - I	AEC	2	0	0	2	50	0	50
Univ Code	Indian Knowledge System	IKS	2	0	0	2	50	0	50
10935401	Professional Ethics	VEC	2	0	0	2	50	50	0
Univ Code	* Co-Curricular Course	CC	2	0	0	2	50	50	0
						22	**550**	**300**	**250**
	*** Co-Curricular Course (Health & Wellness, Yoga education, sports & fitness, Cultural activities, NSS, NCC and Fine/applied/visual/performing arts)**								
	Semester II		**L**	**T**	**P**	**Credits**	**Marks**	**Int**	**Ext**
20035411	Data Structures	Major(Core)	2	0	4	4	100	50	50
20035412	Mathematics Foundations to Computer Science - II	Major(Core)	2	0	0	2	50	0	50
20335411	Programming with Python	Minor Stream	1	0	2	2	50	0	50
20435411	Introduction to Open-Source Technology	OEC	2	0	4	4	100	50	50
20635411	Operating Systems	VSC	1	0	2	2	50	0	50
20735401	Object Oriented Programming	SEC	1	0	2	2	50	50	0

	using Java								
Univ Code	General English - II	AEC	2	0	0	2	50	50	0
20935411	Environmental Studies and Sustainability	VEC	2	0	0	2	50	0	50
Univ Code	* Co-Curricular Course	CC	2	0	0	2	50	50	0
						22	**550**	**250**	**300**
	*** Co-Curricular Course (Health & Wellness, Yoga education, sports & fitness, Cultural activities, NSS, NCC and Fine/applied/visual/performing arts)**								

Distribution of Credits:

1 Hr. Lecture (L) per week	Credit
1 Hr. Tutorial (T) per week	1 Credit
1 Hr. Practical (P) per week	0.5 Credit
2 Hours Practical (P) per week	1 Credit

Course Syllabus Semester I
1.1 Major (Core)

Course Title	**Problem Solving Techniques**
Course Credits	**4 Credits**
	CO1: Familiarize themselves with basic terminology of computers, demonstrate knowledge of problem-solving concepts, describe programming languages and their evolution.
	CO2: Create specification from problem requirements by asking questions to disambiguate the requirement statement.
	CO3: Design the solution from specification of a problem and write pseudo code of the algorithm using basic building blocks or structured programming constructs (Sequence, Selection and Repetition statement).
	CO4: Translate an algorithm into a C computer program
	CO5: Testing and analyzing programs using debugging tools.
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	Analyze complex problems, break them into smaller, manageable subproblems, and define clear input, output, and validation requirements
	Develop algorithms tailored to different types of computational problems, employing appropriate data structures and solution techniques.
	Verify the correctness of algorithms by establishing and testing preconditions and postconditions.
Content Outline	Problems And Problem Instances, Generalization and Special Cases, Types of Computational Problems, Classification of Problems, Analysis of Problems, Solution Approaches, Algorithm Development, Analysis of Algorithm, Efficiency, Correctness, Role of Data Structures in Problem Solving, Problem-Solving Steps (Understand the Problem, Plan, Execute, And Review), Breaking the Problem into Subproblems, Input/Output Specification, Input Validation, Pre and Post Conditions.
Module 2 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	Develop a solid understanding of control structures, including sequence, selection, and repetitio
	Design structured algorithms and represent them clearly using pseudocode and flowcharts.
	Translate algorithms and pseudocode into C programs, using core language elements like scanf(), printf(), operators, control structures, and data types

Content Outline	**Structured Programming Concepts:** Sequence (Input/Output/Assignment), Selection (If, If-Else) And Repetition (For, While, Do-While) Statements, Control Structure Stacking and Nesting. **Different Kinds of Repetitions:** Entry Controlled, Exit Controlled, CounterControlled, Definite, Indefinite and Sentinel-Controlled Repetitions. Pseudocode and Flowcharts. Definition And **Characteristics of Algorithms, Standard Algorithm Format.** Problems Involving Iteration and Nesting: Displaying Different Patterns and Shapes Using Symbols and Numbers, Generating Arithmetic and Geometric Progression, Fibonacci and Other Sequences, Approximate Values For π, Sin(x), Cos(x), Etc. Using Taylor Series. Different Kinds of Data in The Real World and How They are Represented in The Computer Memory. **Representation of Integers:** Signed Magnitude Form, 1's Complement And 2's Complement. Representation of Real Numbers: IEEE 754 Floating Point Representation. Representation of Characters: ASCII, UNICODE. C Language: Introduction To Programming Languages, Different Generations of Programming Languages. Typed Vs Typeless Programming Languages, History of C Language, An Empty C Program. C Language Counterparts For Input (scanf()), Output (printf()) Statements, Assignment, Arithmetic, Relational and Logical Operators. If, If-Else Statements, For, While, Do-While Statements. Data Types. Translating Pseudocode/Algorithm to C Program. Incremental Compilation and Testing of The C Program. Simple Problems Involving Input, Output, Assignment Statement, Selection and Repetition. Good Coding Practices.
Module 3 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	Apply logical reasoning and mathematical operations to solve a variety of number-based problems, such as extracting digits, identifying palindromes, calculating prime factors, detecting special numbers and converting numbers between bases
	Use advanced control structures in C, including the else-if ladder and switch cases, and employ increment/decrement operators, along with break and continue statements.
	Perform statistical operations

Content Outline	**Problems on Numbers:** Extracting Digits of a Number (Left to Right and Right to Left), Palindrome, Prime Number, Prime Factors, Amicable Number, Perfect Number, Armstrong Number, Factorial, Converting Number from One Base to Another. Statistics (Maximum, Minimum, Sum and Average) on a Sequence of Numbers which are Read using Sentinel- Controlled Repetition using only a few Variables. **C Language**: else-if Ladder, switch Case, Increment/Decrement Operators, break and continue Statements, structure, pointers.
Module 4 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	Develop modular programs using both top-down and bottom-up approaches
	Apply recursive functions to solve complex problems, gaining an understanding of when recursion is appropriate.
Content Outline	Modular Programming, Top-Down and Bottom-Up Approaches to Problem Solving. Recursion. Problems on Arrays: Reading and Writing of Array Elements, Maximum, Minimum, Sum, Average, Median and Mode. Sequential And Binary Search. Any one Sorting Algorithm. Matrix Operations. **C Language:** Function Definition and Declaration (Prototype), Role of Return Statement, One Dimensional and Two-Dimensional Arrays. String Functions. Other Operators, Operator Precedence and Associativity. Debugging.

Textbooks

1. Venkatesh, & Nagaraju, Y. (2024). *Practical C Programming for Problem Solving*. Khanna Book Publishing Company.
2. AICTE. (2024). *Programming for Problem Solving (with Lab Manual)*. Khanna Book Publishing Company.
3. Deitel, H., & Deitel, P. (2015). *C How to Program* (9th ed.). Pearson India.
4. Dromey, R. G. *How to Solve It by Computer*.

Reference Books

1. Kernighan, B. W., & Ritchie, D. (2015). *The C Programming Language* (2nd ed.). Pearson.
2. Hanly, J., & Koffman, E. (2015). *Problem Solving and Program Design in C* (8th ed.). Pearson.

Assessment

External Assessment – 50 Marks

Internal Assessment – 50 Marks

Internal Assessment to be done on the basis of Project Presentations and Report Writing with proper Rubrics (Given below are two sample projects but it is expected to work on similar sort of projects)

Project 1:

1) Using C Language create a system that analyzes and evaluates the performance of students based on their scores, using computational problem-solving techniques.

Project 2:

2) Using C Language build a project that performs various matrix operations and demonstrates their real-world applications.

Course Syllabus Semester I

1.2 Major (Core)

Course Title	**Mathematics Foundations to Computer Science - I**
Course Credits	**2 Credits**
	CO1: Familiarize with computer terminology, demonstrate problem-solving skills, and describe programming languages and their evolution."
	CO2: This course enables the students to use mathematical models and techniques to analyze and understand problems in computer science
	CO3: This course demonstrates how the mathematical principles give succinct abstraction of computer science problems and help them to efficiently analyze
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	Understanding and Applying Set Theory and Relations
	Analyzing and Working with Functions
	Mastering Counting Techniques and Recurrence Relations
Content Outline	**Set, Relation and Function:** Set, Set Operations, Properties of Set operations, Subset, Venn Diagrams, Cartesian Products. Relations on a Set, Properties of Relations, Representing Relations using matrices and digraphs, Types of Relations, Equivalence Relation, Equivalence relation and partition on set, Closures of Relations, Warshall's algorithm. Functions, properties of functions (domain, range), composition of functions, surjective (onto), injective (one-to-one) and bijective functions, inverse of functions. Some useful functions for Computer Science: Exponential and Logarithmic functions, Polynomial functions, Ceiling and Floor functions. **Counting and Recurrence Relation:** Basics of counting, Pigeonhole principle, permutation, combination, Binomial coefficients, Binomial theorem. Recurrence relations, modelling recurrence relations with examples, like Fibonacci numbers, the tower of Hanoi problem. Solving linear recurrence relation with constant coefficients using characteristic equation roots method.
Module 2 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	Understanding Fundamental Concepts in Graph Theory

Analyzing and Applying Matrix Operations
Exploring Advanced Matrix Concepts and Application

Content Outline	**Elementary Graph Theory:** Basic terminologies of graphs, connected and disconnected graphs, subgraph, paths and cycles, complete graphs, digraphs, weighted graphs, Euler and Hamiltonian graphs. Trees, properties of trees, concept of spanning tree. Planar graphs. Definitions and basic results on the topics mentioned. **Matrix Algebra:** Types of matrices, algebra of matrices–addition, subtraction, and multiplication of matrices, determinant of a matrix, symmetric and skew-symmetric matrices, orthogonal matrix, rank of a matrix, inverse of a matrix, applications of matrices to solve system of linear equations, Eigen values and Eigen vectors, Caley-Hamilton theorem

Text Books:-

1. Garg, R. (2024). *Engineering Mathematics*. Khanna Book Publishing Company. (AICTE Recommended Textbook)
2. Garg, R. (2023). *Advanced Engineering Mathematics*. Khanna Book Publishing Company.
3. Kolman, B., Busby, R., & Ross, S. (2015). *Discrete Mathematical Structures* (6th ed.). Pearson Education.
4. Deo, N. (1979). *Graph Theory with Application to Engineering and Computer Science*. Prentice Hall, India.
5. Vasishtha, A. R., & Vasishtha, A. K. (2022). *Matrices*. Krishna Prakashan.

Reference Books

1. Grimaldi, R. P., & Ramana, B. V. (2007). *Discrete and Combinatorial Mathematics: An Applied Introduction* (5th ed.). Pearson Education.
2. Rosen, K. H., & Krithivasan, K. (2019). *Discrete Mathematics and its Applications*. McGraw Hill, India.
3. West, D. B. (2015). *Introduction to Graph Theory* (2nd ed.). Pearson Education.

Web Resources

1. https://nptel.ac.in/courses/106103205
2. https://nptel.ac.in/courses/111101115

Assessment

External Assessment – 50 Marks

1.3 OEC

Course Title	**Digital Marketing**
Course Credits	**4 Credits**
Course Outcomes	**After Completion of this Course, students will be able**
	CO1: Demonstrate knowledge of digital marketing fundamentals and their application in the modern marketing mix.
	CO2: Develop skills in creating and managing digital marketing strategies using various tools and platforms.
	CO3: Implement effective online campaigns through SEO, social media, email marketing, and other digital channels.
	CO4: Analyze key performance metrics and optimize marketing strategies for better results
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	Demonstrate the Fundamentals and Scope of Digital Marketing
	Developing and Implementing a Digital Marketing Strategy
	Setting and Measuring Digital Marketing Goals and Performance
Content Outline	**Introduction to Digital Marketing:** Definition and scope of digital marketing, Importance and growth of digital marketing in modern business, Comparison between traditional and digital marketing, The digital marketing ecosystem (Search engines, social media, email marketing, etc.), Key terminologies (CTR, CPC, CPA, ROI, etc.) **Digital Marketing Strategy and Planning:** Developing a digital marketing strategy, understanding buyer personas and target audience, building a customer journey and funnel, Importance of content in digital marketing (content marketing), Integration of digital channels (SEO, social media, Email, PPC), Setting SMART goals and KPIs for digital marketing

Module 2 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	Demonstrate SEO Principles and Website Optimization
	Develop Effective Social Media Marketing Strategies
	Measure and Analyze Marketing Performance:

Content Outline	**Website Analysis and SEO Basics:** Introduction to SEO (Search Engine Optimization), Understanding website structure and user experience, Tools for website analysis (Google Analytics, SEMrush, etc.), Keyword research and on-page optimization **Social Media Marketing:** Overview of social media platforms for marketing (Facebook, Instagram, LinkedIn, Twitter, etc.), Creating engaging content for social media, social media calendar and post scheduling, Basics of paid social media advertising (Facebook Ads, Instagram Ads)
Module 3 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	Demonstrate Email Marketing Fundamentals and List Building
	Design Effective Email Campaigns and Template
	Implement Marketing Automation and Performance Analysis
Content Outline	**Email Marketing and Campaign Automation:** Fundamentals of email marketing, building email lists and segmentation, designing effective email templates, Introduction to email marketing automation (Mailchimp, Active Campaign), Analyzing email marketing performance (Open rates, click-through rates)
Module 4 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	Apply Performance Marketing Concepts and Paid Campaigns
	Analyze Key Digital Marketing Metrics
	Utilize Google Analytics for Campaign Performance Measurement
Content Outline	**Performance Marketing and Analytics:** Introduction to Google Ads and paid search campaigns, Introduction to performance marketing (PPC, affiliate marketing), Understanding key digital marketing metrics and reports, Using Google Analytics for tracking and measuring campaign performance

Textbooks & Reference Books

1. Bhatia, P. S. (2017). *Fundamentals of Digital Marketing*. Pearson Education India.
2. Chaffey, D., & Ellis-Chadwick, F. (2020). *Digital Marketing: Strategy, Implementation, and Practice* (Indian ed.). Pearson Education India.
3. Gupta, S. (2017). *Digital Marketing: Text and Cases*. McGraw Hill Education India.
4. Ahuja, V. (2015). *Digital Marketing: A Practical Approach* (2nd ed.). Oxford University Press India.

5. Dodson, I. (2016). *The Art of Digital Marketing: The Definitive Guide to Creating Strategic, Targeted, and Measurable Online Campaigns*. Wiley India.
6. Kaur, B. (2020). *Introduction to Digital Marketing*. BPB Publications.
7. Das, A. (2020). *Digital Marketing: Concepts, Strategy and Practice*. Himalaya Publishing House.
8. Mehta, R. (2019). *SEO & Social Media Marketing: A Comprehensive Guide to Optimize Websites, Increase Traffic, and Engage Users*. Rohit Mehta Publishing.

Assessment
External Assessment – 50 Marks
Internal Assessment – 50 Marks

Internal Assessment to be done on the basis of Project Presentations and Report Writing with proper Rubrics (Given below are two sample projects but it is expected to work on similar sort of projects)

Project 1: Design and execute a social media campaign to promote a product, service, or event.

(Choose a target audience and platform (e.g., Instagram, LinkedIn, Facebook, create content (images, videos, carousels), Schedule posts using tools like Hootsuite or Buffer, Analyse campaign performance using insights from the platform.)

Project 2: Create and manage a Google Ads campaign for a specific goal (e.g., increasing website traffic or conversions).
(Perform keyword research using Google Keyword Planner, write ad copy and set up ad groups,
Define budgets, bids, and targeting options, Track and analyse ad performance using Google Ads analytics)

1.4 VSC

Course Title	**Computer Architecture**
Course Credits	**2 Credits**
Course Outcomes	**After Completion of this Course, students will be able**
	CO1: Familiarize with the fundamentals of Digital Electronics and Binary Number Systems
	CO2: To Learn the implementation of Combinational Circuit
	CO3: To Learn the implementation of Sequential Circuit
	CO4: Comprehend the basic principles of computer organization and design
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	Demonstrate Digital Signals and Computer Architecture
	Apply Boolean Algebra and Karnaugh Maps for Simplification
	Convert and Performing Arithmetic Operations in Various Number Systems
Content Outline	**Digital Principles**: Introduction to Digital Signals, Definition and characteristics of digital signals vs. analog signals, Basic understanding of digital logic and digital computers. **Von Neumann Architecture**: Overview of Von Neumann architecture and its significance in digital systems. **Boolean Laws and Theorems**: Introduction to basic Boolean algebra, K- Map (Karnaugh Map): Focus on 2 and 3-variable K-Maps for simplification, Simplification of Boolean expressions using SOP (Sum of Products) and POS (Product of Sums) forms, Don't Care Conditions for optimization of K-Maps (at a basic level). **Basic Number Systems:** Overview of Decimal, Binary, Octal, and Hexadecimal number systems, Basic Number System Conversions between binary, decimal, octal, and hexadecimal. **Binary Arithmetic:** Simple binary arithmetic (addition and subtraction). Binary and Decimal Codes: **ASCII and Gray Code basics, Brief introduction to Error-detecting codes.**
Module 2 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	Apply Combinational Circuit Design
	Explore Sequential Circuits and Flip-Flop
	Utilize Shift Registers and Binary Counter

Content Outline	**Combinational and Sequential Circuits:** Half Adder and Full Adder: Basic design and working, Multiplexer and Demultiplexer: Overview and practical applications. **Sequential Circuits:** Flip-Flops: SR Flip-Flop and D Flip-Flop. **Shift Registers:** Introduction to 4-bit Shift Registers and their practical uses. Binary Counters: I**ntroduction to a 4-bit synchronous binary counter.**

Textbooks

1) Leach, D. P., Malvino, A. P., & Saha, G. (2011). *Digital Principles and Applications* (2011 ed.). Tata McGraw Hill Education Private Limited.
2) Mano, M. M. (2007). *Computer System Architecture* (3rd ed.). Pearson/PHI.

Reference Books

3) Stallings, W. (2003). *Computer Organization and Architecture* (6th ed.). Pearson/PHI.
4) Tanenbaum, A. S. (1999). *Structured Computer Organization* (4th ed.). PHI/Pearson.
5) Subramanyam, M. V. (2010). *Switching Theory and Logic Design*. Laxmi Publications (P) Ltd.
6) Singh, I. (2020). *Computer Organization and Architecture*. Khanna Book Publishing.

Assessment

Internal Assessment – 50 Marks

Internal Assessment to be done on the basis of Project Presentations and Report Writing with proper Rubrics (Given below are two sample projects but it is expected to work on similar sort of projects)

Project 1: Using simulation tool Logisim Build and test a Half Adder and Full Adder

Project 2: Using simulation tool Logisim design a 4-bit ALU that can perform multiple arithmetic and logical operations.

1.5 SEC

Course Title	**Web Technology - I**
Course Credits	**2 Credits**
Course Outcomes	**After Completion of this Course, students will be able**
	CO1: Familiarize with the basics of web technologies and the Internet
	CO2: Develop skills in designing static and dynamic web pages using HTML and CSS.
	CO3: Develop proficiency in creating interactive forms and multimedia content for web applications.
	CO4: Gain proficiency in crafting responsive, user-friendly, and visually appealing web designs.
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	Describe the Internet and Web Technologies
	Apply Web Standards and Best Practices
Content Outline	**Introduction to Web Technologies Overview of the Internet** History and evolution of the web. Understanding web protocols (HTTP, HTTPS, FTP). The role of web browsers and servers. **Web Standards** W3C guidelines and web accessibility (WCAG). Importance of semantic web and SEO. **HTML5** Structure of an HTML document. Key HTML elements: headings, paragraphs, lists, links, images, tables, and forms. Semantic elements (<header>, <nav>, <section>, <article>, <footer>, etc.). Multimedia elements: audio and video tags. HTML5 form enhancements: new input types and attributes (e.g., date, email, required, etc.)
Module 2 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	Apply the Basics and styles of CSS
	Differentiate Between Types of CSS and Their Uses
	Utilize CSS Selectors, properties for Styling

Content Outline	**Cascading Style Sheets (CSS)** **Introduction to CSS (5 Hours) Overview of CSS** History and evolution of CSS. Importance of CSS in web development. CSS syntax: selectors, properties, and values. **Types of CSS** Inline CSS: usage and limitations. Internal CSS: defining styles within <style> tags. External CSS: linking external stylesheets using <link> tags. **CSS Selectors** Basic selectors: element, class, and ID selectors. Grouping and combining selectors: descendant, child, adjacent sibling, general sibling selectors. Pseudo-classes and pseudo-elements: hover, focus, before, after T**ext Properties** Font properties: font-family, font-size, font-weight, font-style. Text alignment, spacing, and decoration: text-align, line-height, letter-spacing, text-decoration. **Colour and Backgrounds** Colour models: HEX, RGB, RGBA, HSL, HSLA. Background properties: background-colour, background-image, background-size, background-repeat, background-position.

Textbooks

1. Kumar, R. (2012). *Web Technologies: Theory and Practice*. Dreamtech Press.
2. Patel, K. (2015). *Fundamentals of Web Development*. PHI Learning.
3. Mohan, P. K. (2016). *Web Technologies: An Overview*. New Age International.
4. Bansal, N., & Kumar, S. (2018). *HTML, CSS, and JavaScript: A Comprehensive Guide*. Laxmi Publications.

Reference Books

1. Patel, A. (2017). *Responsive Web Design: Techniques and Best Practices*. Oxford University Press.
2. Sharma, R. K., & Kumar, N. (2021). *Fundamentals of Web Programming*. Tata McGraw-Hill Education.

Assessment

Internal Assessment – 50 Marks

Internal Assessment to be done on the basis of Project Presentations and

Report Writing with proper Rubrics (Given below are two sample projects but it is expected to work on similar sort of projects)

Project 1: Using HTML/CSS design a single product page for an e-commerce site.
Project 2: Using HTML/CSS design a single-page resume builder using HTML forms.

1.6 AEC

Course Title	**General English - I**
Course Credits	**2 Credits**
Course Outcomes	**After Completion of this Course, students will be able**
	CO1: To provide learning environment to practice listening, speaking, reading and writing skills
	CO2: To assist the students to carry on the tasks and activities through guided instructions and materials.
	CO3: To effectively integrate English language learning with employability skills and training.
	CO4: To provide hands-on experience through case-studies, mini- projects, group and individual presentations.
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	enhance their vocabulary and understanding of word formation, construct clear and grammatically correct sentences, and develop a precise and organized writing style.
	identifying and correct common writing errors, resulting in more effective and polished written communication.
Content Outline	**Vocabulary Building**: The concept of Word Formation, Root words from foreign languages and their use in English, Acquaintance with prefixes and suffixes from foreign languages in English to form derivatives, Synonyms, antonyms, and standard abbreviations. **Basic Writing Skills:** Sentence Structures, Use of phrases and clauses in sentences, Importance of proper punctuation, creating coherence, organizing principles of paragraphs in documents, Techniques for writing precisely **Identifying Common Errors in Writing:** Subject-verb agreement, Noun-pronoun agreement, Misplaced modifiers, Articles, Prepositions, Redundancies
Module 2 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	Develop the ability to write sensibly and coherently, structuring ideas in various forms of writing.
	Enhance oral communication skills, including listening, pronunciation, intonation, and presentation, preparing them for both informal and professional interactions.
	Gain confidence and clarity in their spoken and written communication, equipping them with essential skills for academic, social, and workplace settings.

Content Outline	**Nature and Style of sensible Writing:** Describing, Defining, Classifying, providing examples or evidence, writing introduction and conclusion, Module V: Writing Practices, Comprehension, Précis Writing, Essay Writing **Oral Communication (This Module involves interactive practice sessions in Language Lab)** Listening Comprehension, Pronunciation, Intonation, Stress and Rhythm, Common Everyday Situations: Conversations and Dialogues, Communication at Workplace, Interviews, Formal Presentations

Text Books & Reference Books

1. Tiwari, A. (2023). *Communication Skills in English (with Lab Manual)*. Khanna Book Publishing Co.
2. Kumar, K. B. (2022). *Effective Communication Skills*. Khanna Book Publishing.
3. Swan, M. (1995). *Practical English Usage*. Oxford University Press.
4. Wood, F. T. (2007). *Remedial English Grammar*. Macmillan.
5. Zinsser, W. (2001). *On Writing Well*. Harper Resource Book.
6. Hamp-Lyons, L., & Heasly, B. (2006). *Study Writing*. Cambridge University Press.
7. Kumar, S., & Lata, P. (2011). *Communication Skills*. Oxford University Press.
8. Central Institute of English and Foreign Languages (CIEFL), Hyderabad. (n.d.). *Exercises in Spoken English (Parts I-III)*. Oxford University Press.

Assessment

External Assessment – 50 Marks

1.7 IKS

Indian Knowledge System (IKS) Syllabus as Per NEP | SNDT Women's University

Assessment
External Assessment – 50 Marks

1.8 VEC

Course Title	**Professional Ethics**
Course Credits	**2 Credits**
Course Outcomes	**After Completion of this Course, students will be able**
	CO1: Familiarize with the fundamental concepts and theories of professional ethics.
	CO2: Apply ethical theories and models to real-world scenarios.
	CO3: Recognize and address ethical dilemmas in professional practice
	CO4: Integrate professional ethics into daily work practices.
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	Articulate the importance of professional ethics in computer science and understand key ethical theories and codes of conduct relevant to the field.
	Identify and critically evaluate ethical issues in technology, particularly those related to privacy, data usage, and software confidentiality.
Content Outline	**Introduction to Professional Ethics in Computer Science** • Definition and Importance of Professional Ethics • Ethical Theories: Utilitarianism, Deontology, Virtue Ethics • Code of Ethics in Computer Science (ACM, IEEE) • Ethical Responsibilities of Computer Professionals Overview of Ethical Issues in Technology **Privacy and Confidentiality** • Understanding Privacy in the Digital Age • Ethical Implications of Data Collection and Usage Confidentiality Issues in Software Development
Module 2 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	Demonstrate a clear understanding of intellectual

property rights and

	apply copyright, trademark, and patent principles to software projects responsibly.
	Apply these ethical principles in decision-making, maintaining a commitment to ethical and responsible practices within both software development and cybersecurity contexts.
Content Outline	**Intellectual Property and Copyright** • Overview of Intellectual Property Rights • Copyright, Trademarks, and Patents in Software • Ethical Use of Open-Source Software Plagiarism in Programming and Software Development **Security, Cybercrime, and Professional Accountability** • Ethical Considerations in Cybersecurity • Responsibility for Security in Software Development Ethical Hacking vs. Malicious Hacking

Textbooks

1. Reynolds, G. (2018). *Ethics in Information Technology* (6th ed.). Cengage Learning.
2. Spinello, R. A. (2007). *Computer Ethics: A Global Perspective*. Jones and Bartlett Publishers.
3. Quinn, M. J. (2024). *Ethics for the Information Age* (9th ed.). Pearson Education.

Reference Books

1. Perkins, C. B. (2001). *Computers and Society*. Pearson Education.
2. Spinello, R. A. (2003). *The Ethics of Information Technology and Business*. Wadsworth Publishing.
3. Cooper, T. L. (2006). *The Responsible Administrator: An Approach to the Ethics of the Profession* (3rd ed.). Jossey-Bass.

Assessment

Internal Assessment – 50 Marks

Internal Assessment to be done on the basis of Project Presentations and Report Writing with proper Rubrics (Given below are two sample projects but it is expected to work on similar sort of projects)

Project1: projects to analyze specific issues, such as ethical hacking or intellectual property laws in software development.

Project2: projects on situations related to intellectual property or cybersecurity and ask for ethical solutions.

Course Syllabus Semester: I

1.9 CC https://sndt.ac.in/nep2020/syllabus-as-per-nep/cc-syllabus

Assessment
Internal Assessment – 50 Marks (as per university syllabus)

Course

Syllabus

Semester: II

2.1 Major (Core)

Course Title	**Data Structures**
Course Credits	**4 credits**
Course Outcomes	**CO1:** Demonstrate Data Structures concepts and their practical applications
	CO2: Apply problem-solving skills using Data Structures.
	CO3: Analyze Data Structures using C programming language.
	CO4: Design graph and tree data structures to solve real-world problems.
Module 1 (Credit 1)	
Learning Outcomes	**Learners will be able to**
	Define Data Structures and describe their role in organizing, managing, and storing data efficiently for various applications.
	Explain Basic Data Structure Operations, including creation, insertion, deletion, updating, and traversal.
	Define Multi-Dimensional Arrays and explain their representation in memory for applications that require higher-dimensional data organization.
	Apply Linear and Binary Search Techniques to search for elements within an array and compare their efficiency.
Content Outline	**Introduction and Overview:** Definition, Classification and Operations of Data Structures. Algorithms: Complexity, Time- Space Tradeoff. **Arrays:** Definition and Classification of Arrays, Representation of Linear Arrays in Memory, Operations on Linear Arrays: Traversing, Inserting, Deleting, Searching, Sorting and Merging. Searching: Linear Search and Binary Search, Comparison of Methods. Sorting: Bubble Sort,Selection Sort, and Insertion Sort. Two-Dimensional Arrays, Representation of Two- Dimensional Arrays in Memory, Matrices and Sparse Matrices, Multi-Dimensional Arrays.
Module 2 (Credit 1)	

Learning Outcomes	Learners will be able to
	Define Linked Lists and explain their structure, highlighting the differences between linked lists and arrays in terms of memory allocation, flexibility, and performance.
	Demonstrate Applications of Linked Lists
	Define Hashing and its importance in data retrieval, explaining the concept of hash functions.
	Explain the Concept of Collision in hashing, discussing the implications of multiple keys hashing to the same index in a hash table.

Content Outline	**Linked Lists:** Definition, Comparison with Arrays, Representation, Types of Linked lists, Traversing, Inserting, Deleting and Searching in Singly Linked List, Doubly Linked List and Circular Linked List. Applications of Linked Lists: Addition of Polynomials. **Hashing and Collision:** Hashing, Hash Tables, Types of Hash Functions, Collision, Collision Resolution with Open Addressing and Chaining.
Module 3 (Credit 1)	**Learners will be able to**
Learning Outcomes	Learn Stacks and explain their Last In, First Out (LIFO) principle, highlighting the real-world analogy of stack data structures and Understand Applications of Stacks
	Learn Recursion and explain its characteristics, including base cases and recursive cases. apply Recursive Notation and how it relates to problem- solving in programming.
	Design Queues and explain their First In, First Out (FIFO) principle, along with real-world examples.
Content Outline	**Stacks**: Definition, Representation of Stacks using Arrays and Linked List, Operations on Stacks using Arrays and Linked List, Application of Stacks: Arithmetic Expressions, Polish Notation, Conversion of Infix Expression to Postfix Expression, Evaluation of Postfix Expression. **Recursion:** Definition, Recursive Notation, Runtime Stack, Applications of Recursion: Factorial of Number, GCD, Fibonacci Series and Towers of Hanoi. **Queues:** Definition, Representation of Queues using Array and Linked List, Types of Queue: Simple Queue, Circular Queue, Double-Ended queue, Priority Queue, Operations on Simple Queues and Circular Queues using Array and Linked List, Applications of Queues**.**
Module 4 (Credit 1)	**Learners will be able to**
Learning Outcomes	Describe Graph Representation Methods
	Implement Graph Traversal Techniques
	Apply Binary Trees as a special type of tree where each node has at most two children, and recognize their properties.

	Design Height Balanced Trees (AVL Trees) as a type of self- balancing binary search tree, maintaining balance through specific height conditions.
Content Outline	**Graphs:** Definition, Terminology, Representation, Traversal. **Trees:** Definition, Terminology, Binary Trees, Traversal of Binary Tree, Binary Search Tree, Inserting, Deleting and Searching in Binary Search Tree, Height Balanced Trees: AVL Trees, Insertion and Deletion in AVL Tree.

Text Books

1. R.B. Patel, "Expert Data Structures with C", Khanna Book Publishing Company, 2023(AICTE Recommended Textbook)
2. Seymour Lipschutz, "Data Structures with C", Schaum's Outlines, Tata McGraw-Hill,2011.
3. Yashavant Kanetkar, "Data Structures Through C", 4th Edition, BPB Publications,2022.

Reference Books

1. Reema Thareja, "Data Structures Using C", Second Edition, Oxford University Press,2014.
2. Ellis Horowitz, Sartaj Sahni, and Susan Anderson-Freed, "Fundamentals of DataStructures in C", Second Edition, Universities Press, 2007.

Web Resources

1. Geeks for Geeks - Data Structures Tutorial
2. Khan Academy - Algorithms Course

Assessment

External Assessment – 50 Marks
Internal Assessment – 50 Marks

Internal Assessment to be done on the basis of Project Presentations and Report Writing with proper Rubrics (Given below are two sample projects but it is expected to work on similar sort of projects)

Project 1:

Using C language create a library management system that handles book borrowing, user registrations, book searches, and reservations
(Use: Hash Tables for fast lookups of books, Linked Lists for maintaining queues of borrowed books, Trees for maintaining ordered catalogues)

Project 2:

Using C language create a flight reservation system where users can book, cancel, and search for flights. (Use: Arrays or Lists for flight schedules, Trees for efficient searching of available flights, Queues for handling waiting lists)

Course

Syllabus

Semester: II

2.2 Major (Core)

Course Title	**Mathematics Foundations to Computer Science - II**
Course Credits	**2 Credits**
Course Outcomes	**CO1:** Develop critical thinking skills to construct and evaluate logical arguments and proofs.
	CO2: This course introduces mathematical techniques that are foundations for understanding advanced computational methods, including numerical methods and optimization.
	CO3: This course helps the students to understand various problem-solving strategies and methods to tackle both theoretical and practical challenges in computer science.
Module 1 (Credit 1)	
	After learning the module, learners will be able to
Learning Outcomes	Analyze and construct logical statements, and evaluate their truth values.
	Construct logical proofs using various methods and justify statements through formal reasoning.
	Recognize and apply algebraic structures like groups in mathematical and practical contexts, understanding their core properties and behaviors.
Content Outline	**UNIT I: Logic and Methods of Proofs:** Propositions, logical operations (basic connectives), compound statements, construction of truth table, quantifiers, conditional statements, tautology, contradiction, contingency, logical equivalence. Conjunctive Normal Forms (CNF) and Disjunctive Normal Forms (DNF). **Methods of proofs**: Rules of inference for propositional logic, modus ponens, modus tollens, syllogism, proof by contradiction, Mathematical Induction. **Algebraic Structures:** Semi-group, Monoid, Group, Subgroup, Cyclic group.
Module 2 (Credit 1)	
	After learning the module, learners will be able to

Learning Outcomes	**Identify and analyze** numerical methods to solve equations, interpolate values, and perform numerical integration.

	Solve problems using prescribed formulas for the Bisection, Newton- Raphson, Interpolation, and Numerical Integration methods.
	Formulate and solve linear programming problems, using both graphical and simplex methods, while understanding special cases and duality.
	Apply transportation methods to optimize transportation costs, understanding various approaches to obtain feasible and optimal solutions.
Content Outline	**UNIT II: Numerical Methods:** Concept and importance of errors in numerical methods. Solution of algebraic and transcendental equations: Bisection method and Newton-Raphson methods. **Numerical Interpolation**: Newton's Forward and Newton's Backward interpolation formula and Lagrange's formula. Numerical Integration: Trapezoidal rule and Simpson's 1/3 rule Only formula and problem solving for all the topics mentioned above. **Linear programming:** Introduction, LP formulation, Graphical method for solving LPs with two variables, Special cases in graphical methods, Simplex method, Duality. **Transportation problem**: Definition, Linear form, North-west corner method, least cost method, Vogel's approximation method for finding feasible solution, MODI method for finding optimum solution.

Textbooks

1. Kolman, B., Busby, R., & Ross, S. (2015). *Discrete Mathematical Structures* (6th ed.). Pearson Education.
2. Sastry, S. S. (2022). *Introductory Methods of Numerical Analysis* (5th ed.). PHI Learning.
3. Taha, H. A. (2003). *Operations Research: An Introduction* (8th ed.). Pearson Prentice Hall.
4. Singh, S. B. (2023). *Discrete Structures*. Khanna Book Publishing.

Reference Books

1. Rosen, K. H., & Krithivasan, K. (2019). *Discrete Mathematics and its Applications*. McGraw Hill Education India.

Web Resources

1. https://nptel.ac.in/courses/111107127
2. https://www.math.iitb.ac.in/~siva/si50716/SI507lecturenotes.pdf

Assessment
External Assessment – 50 Marks

Course

Syllabus

Semester: II

2.3 Minor Stream

Course Title	**Programming with Python**
Course Credits	**2 Credits**
Course Outcomes	**CO1**: Gain proficiency in the basics of Python programming
	CO2: Develop problem-solving skills using programming.
	CO3: Acquire skills in programming fundamentals, including variable declaration, looping structures, and function implementation.
	CO4: Design, debug, and evaluate Python programs to ensure functionality and reliability.
Module 1 (Credit 1)	
Learning Outcomes	**Learners will be able to:**
	Demonstrate Python's environment and syntax.
	Learn how to declare and use variables effectively.
	Implement decision-making and repetitive structures in Python.
	Develop problem-solving logic using control flow constructs.
Content Outline	**Introduction to Python** • Overview of Python: Features, uses, and installation. • Writing and executing Python programs. • Basic syntax: Indentation, comments, input/output, and variables. • Data types: Integers, floats, strings, and booleans. • Type casting and basic operators (arithmetic, relational, logical). **Control Flow** • Conditional Statements: if, if-else, and nested conditions. • Looping Constructs: for, while, and else with loops. • Break, continue, and pass statements.
Module 2 (Credit 1)	
Learning Outcomes	**Learners will be able to:**
	Write reusable code using functions.
	Apply modular programming techniques
	Identify and fix errors in Python programs.
	Implement robust programs with proper error handling.

Content Outline	**Functions and Modular Programming** • Defining and calling functions. • Function arguments and return values. • Scope and lifetime of variables. • Lambda functions. • Importing and using modules (math, random, etc.) **Debugging and Error Handling** • Types of errors: Syntax, runtime, and logical errors. • Debugging techniques and tools. • Exception handling: try, except, finally, and raising exceptions.

Textbooks:

1. Kamthane, A. N., & Kamthane, A. A. (2018). Programming and problem solving with Python. Pearson Education India.
2. Rao, R. N. (2018). Core Python programming. Dreamtech Press.
3. Taneja, S., & Kumar, N. (2017). Python programming: A modular approach. Pearson Education India.
4. Thareja, R. (2023). Python programming. Oxford University Press India.
5. Kanetkar, Y. (2019). Let us Python. BPB Publications.
6. Jain, S. (2020). O Level Programming and Problem Solving Through Python Language: Made Simple. BPB Publications.

Assessment

External Assessment – 50 Marks

Course Syllabus Semester: II

2.4 OEC

Course Title	**Introduction to Open-Source Technology**
Course Credits	**4 Credits**
Course Outcomes	**CO1:** Explore the concepts, benefits, and ethics of open-source software development.
	CO2: Equip students with practical skills in open-source tools on Windows systems.
	CO3: Demonstrate the role and significance of open-source software in modern technology.
	CO4: Build foundational knowledge for future exploration and careers in open-source ecosystems.
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	Explain the principles, history, and significance of open-source technologies.
	Differentiate between various open-source licenses and apply the appropriate license in real-world scenarios.
Content Outline	**Introduction to Open-Source Philosophy and Basics** Definition and core principles of open source. - Overview of open-source licenses: GPL, MIT, Apache, BSD. - Philosophy of open source: community-driven development, free software movement, open innovation. - Case studies: Linux, VLC, Mozilla Firefox. - Benefits of open source: cost-effectiveness, customizability, security, and reliability. - Impact of open-source technologies in industries like AI, cloud computing, and cybersecurity.
Module 2 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	Use Git for version control effectively, including branching, merging, and conflict resolution.

	Explore and utilize open-source tools for productivity and collaboration.
Content Outline	**Version Control and Collaboration: -** - Importance of version control in collaborative software development. - Introduction to Git: initialization, key commands (add, commit, push, pull, branch, merge). - Collaboration workflows: forking, pull requests, resolving merge conflicts. - Using Git hosting platforms: GitHub repositories, issues, and code reviews.
Module 3 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	Utilize open-source Integrated Development Environments (IDEs)
	Apply collaborative open-source project management tools,
Content Outline	**Open-Source Tools for Productivity** - Alternatives to proprietary software: document creation (LibreOffice), image editing (GIMP, Inkscape). - Tools for software development: IDEs like Visual Studio Code. - Collaborative tools: open-source project management tools like Trello or Open Project.
Module 4 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	Utilize open-source tools for cloud computing and containerization.
	Review and create pull requests in open-source repositories
	evaluate the advantages and challenges
Content Outline	**Exploring Open-Source Applications:** - Real-world applications: operating systems (Linux), web browsers (Firefox, Chromium), media players (VLC). - Open-source tools for cloud computing and containerization (Docker). - Introduction to contributing to open-source communities: understanding repositories, filing issues, reviewing pull requests.

Textbooks:

1. Weber, S. (2004). Open-source software: A creative commons approach. MIT Press.
2. Chacon, S., & Straub, B. (2014). Git pro. Apress.
3. Raymond, E. S. (1999). The cathedral & the bazaar. O'Reilly Media.
4. Fogel, K. (2005). Producing open-source software: How to run a successful free software project. O'Reilly Media.
5. Negus, C. (2013). Linux bible (8th ed.). Wiley.

References:

1. Loeliger, J., & McCullough, M. (2012). Version control with git (2nd ed.). O'Reilly Media.
2. Kylander, K., & Kylander, O. S. (2017). GIMP: The official handbook. GIMP Documentation Team.
3. LibreOffice Documentation Team. (2018). LibreOffice 6.0 writer guide. LibreOffice Documentation Team.

Web Resources

1) Git Documentation: https://git-scm.com/doc
2) LibreOffice: https://www.libreoffice.org/
3) GIMP: https://www.gimp.org/
4) VLC Media Player: https://www.videolan.org/

Assessment

External Assessment – 50 Marks
Internal Assessment – 50 Marks

Internal Assessment to be done on the basis of Project Presentations and Report Writing with proper Rubrics (Given below are two sample projects but it is expected to work on similar sort of projects)

Project 1: Using IDEs like Visual Studio Code Develop an open-source tool for enhancing productivity (e.g., a task management app, note-taking tool, or simple text editor) and share it on GitHub.

Project 2: Using IDEs like Visual Studio Code develop an open-source web application (e.g., a personal blog, portfolio, or task manager) using open-source technologies and deploy it on a cloud platform.

Course Syllabus Semester: II

2.5 VSC

Course Title	**Operating Systems**
Course Credits	2 Credits
Course Outcomes	**CO1:** Define the fundamental characteristics and features of operating systems
	CO2: Comprehend multithreaded programming, CPU scheduling, process management, process synchronization, memory, deadlocks, and storage management.
	CO3: Compare the performance of CPU scheduling algorithms
	CO4: Identify the features of I/O and File handling methods.
Module 1 (Credit 1)	
Learning Outcomes	**Learners will be able to:**
	Use Operating Systems and Their Components
	Identifying the Process Management and Multithreading Concepts
	Apply Grasping Process Management and Multithreading Concepts
Content Outline	**UNIT I:** **Operating Systems Overview:** Definition, Components & Services of OS, Structure, Architecture, types of Operating Systems, Batch Systems, Concepts of Multiprogramming and Time Sharing, Parallel, Distributed and real time Systems. **Operating Systems Structures**: Operating system services and systems calls, system programs, operating system structure. **Process Management**: Process Definition, Process states, Process State transitions, Process Scheduling, Process Control Block, Threads, Concept of multithreads, Benefits of threads, Types of threads. **Process Scheduling**: Definition, Scheduling objectives, Scheduling algorithms, CPU scheduling Preemptive and Non-preemptive Scheduling algorithms (FCFS, SJF and RR), Performance evaluation of the scheduling Algorithms.

Module 2 (Credit 1)	
Learning Outcomes	**Learners will be able to:**
	Understanding Process Synchronization and Deadlock Management
	Managing Memory and Virtual Memory
	Applying I/O Management and Disk Scheduling Techniques
Content Outline	**UNIT II:** **Process Synchronization**: Introduction, Inter-process

	Communication, Race Conditions, Critical Section Problem, Mutual Exclusion, Semaphores, Monitors. **Deadlocks**: System model, deadlock characterization, deadlock prevention, avoidance, Banker's algorithm, Deadlock detection, and recovery from deadlocks. **Memory Management**: Logical and Physical address map, Swapping, Memory allocation, MFT, MVT, Internal and External fragmentation and Compaction, Paging, Segmentation. **Virtual Memory:** Demand paging, Page Replacement algorithms, Allocation of frames, thrashing. **I/O Management**: Principles of I/O Hardware: Disk structure, Disk scheduling algorithms.

Textbooks:

1. Walia, E. (2022). *Operating systems concepts*. Khanna Publishing House.
2. Silberschatz, A., Galvin, P. B., & Gagne, G. (2006). *Operating system principles* (7th ed.). Wiley India Pvt. Ltd.
3. Stallings, W. (2006). *Operating systems: Internals and design principles* (5th ed.). Pearson Education India.

Reference Books:

1. Tanenbaum, A. S. (2007). *Modern operating systems* (3rd ed.). Prentice Hall India.

Assessment

External Assessment – 50 Marks

Course
Syllabus
Semester: II
2.6 SEC
Object Oriented Programming using Java

Course Title	**Object Oriented Programming using Java**
Course Credits	**2 Credits**
Course Outcomes	**CO1:** To introduce the object-oriented programming system concepts
	CO2: To introduce syntax and semantics of Java programming language
	CO3: To develop modular programs using Java
	CO4: To setup JDK environment to create, debug and run Java programs
Module 1 (Credit 1)	
Learning Outcomes	**Learner will be able to:**
	Apply Object-Oriented Programming Concepts
	Demonstrate Proficiency with Java Syntax and Language Fundamentals
	Implement Control Flow and Decision-Making in Java Programs
Content Outline	**UNIT I:** **Fundamentals of Object-Oriented Programming**: Basic Concepts of Object-Oriented Programming (OOP), Benefits and Applications of OOP. **Java Evolution**: Java Features, Difference between Java, C and C++, Java and Internet, Java Environment. **Overview of Java Language**: Introduction to Simple Java Program, Use of Comments and Math function, Application of two classes, Java Program Structure, Java Tokens and statements, Implementing Java program And JVM, Command Line Arguments. **Constants, Variables and Data Types:** Constants, Variables, Data Types, Declaration of Variables, Giving values to Variables, Symbolic Constants, Typecasting. **Operators & Expressions**: Arithmetic operators, Relational operators, Logical operators, Assignment operators, Increment & Decrement operators, conditional operators, Bitwise operators, Arithmetic Expressions, Evaluation of Expressions, Type Conversions in Expressions, Operator Precedence & Associativity. Decision Making, Branching & Looping: Decision Making with Control Statements, Looping statements, Jump in loops, Labelled

	loops.
Module 2 (Credit 1)	
Learning Outcomes	**Learner will be able to:**
	Develop Proficiency in Object-Oriented Programming Concepts:
	Effectively Utilize Arrays, Strings, and Collections:
	Apply Inheritance, Interfaces, and Exception Handling

Content Outline	**UNIT II:** Classes, Objects and Methods: Defining Class, Methods Declaration, Constructors, Methods Overloading, Overriding Methods, Inheritance Arrays, Strings and Vectors: 1D arrays, Creating an Array, 2D arrays, Strings, Vectors, Wrapper Classes, Enumerated Types Inheritance: Defining, extending classes, and Implementing Interfaces. Multiple inheritance and polymorphism. **Packages**: Basics of packages, System packages, Creating and accessing packages, creating user defined packages, Adding class to a package. **Exception Handling:** Using the main keywords of exception handling: try, catch, throw, throws and finally; Nested try, Multiple catch statements, Creating user defined exceptions.

Textbooks:

1. Balaguruswamy, E. (2023). Programming with Java: A Primer (7th ed.). McGraw Hill Education.
2. Schildt, H. (2022). Java: The Complete Reference (12th ed.). McGraw-Hill Education.

Reference Books:

1. Goyal, A. (2012). The Essentials of JAVA. Khanna Book Publishing Company Private Limited.
2. Alam, T. (2015). Core JAVA. Khanna Book Publishing Company Private Limited.
3. Liang, Y. D. (2008). Introduction to Java Programming (7th ed.). Pearson.
4. Malhotra, S., & Choudhary, S. (2014). Programming in Java (2nd ed.). Oxford University Press.

Web Resources

1. https://www.w3schools.com/java/.
2. http://www.java2s.com/.
3. https://onlinecourses.nptel.ac.in/noc22_cs47/preview

Assessment

Internal Assessment – 50 Marks

Internal Assessment to be done on the basis of Project Presentations and Report Writing with proper Rubrics (Given below are two sample projects but it is expected to work on similar sort of projects)

Project 1: Using jdk design and implement a student grading system that evaluates students based on their marks and provides functionality for

calculating grades, generating transcripts, and displaying student information.

Project 2: Using jdk design a Bank Account System which allows users to create accounts, deposit money, withdraw money, and check balances.

Course

Syllabus

Semester: II

2.7 AEC

Course Title	**General English - II**
Course Credits	**2 Credits**
Course Outcomes	**CO1:** Students will be able to draft effective business correspondence with brevity and clarity.
	CO 2: Enhance critical thinking by designing and developing clean and lucid writing skills
Module 1 (Credit 1) Fundamentals of Communication	
Learning Outcomes	**Learners will be able to**
	Demonstrate the fundamental concepts of interpersonal and professional communication.
	Learn how to use grammar rules and vocabulary in real-life contexts, enhancing their language proficiency
Content Outline	Basics of Communication, Process of Communication, Components of Communication, Factors of Communication, Barriers to Communication – Physical, Psychological, Semantics, Organizational and Interpersonal Barriers; How to overcome Barriers. Verbal (Written & Oral), Non-verbal - Non-Verbal Communication -Personal appearance; Facial Expression, Movement, Posture, Gesture, Eye Contact. Parts of speech, Verb, Tenses: Form and use, Articles and Prepositions, Transformations of sentences, Common Errors. Root words (Etymology), Meaning of Words in Context, Synonyms & Antonyms, Collocations, Prefixes & Suffixes, Standard Abbreviations.
Module 2 (Credit 1) Grammar in context and Vocabulary Building and Kinesics – Voice Dynamics	
Learning Outcomes	**Learners will be able to**
	Produce a variety of communication materials and technical documents that meet professional standards
	Interpret and utilize kinesics effectively, as well as develop vocal techniques to convey meaning
Content Outline	Report Writing, Parts of a Formal Letter and Formats Parts/Elements of a Formal Letter - Letter heads, and/or Sender's Address, Dateline, Inside Address, Reference. Basics of Email Types of Letters in Both Formal Letter Format and Emails Claim & Adjustment Letters, Request/Permission Letters Emails- Format of Emails,

	Features of Effective Emails, Language and style of emails. Kinesics: Definitions; importance; Features of Body Language; Voice Modulation: Quality, Pitch; Rhythm; intonation; Pronunciation; Articulation; stress & accent; Linguistic features of voice control: Vowel & Consonant Sounds.

Textbooks:

1. **Rai, U., & Rai, S. M. (2004). *Business communications*. Himalaya Publishing House.**
2. **Sharma, R. C., & Mohan, K. (2001). *Business correspondence and report writing*. Tata McGraw Hill & Co. Ltd.**

Assessment

Internal Assessment – 50 Marks

Internal Assessment to be done on the basis of Project Presentations and Report Writing with proper Rubrics (Given below are two sample projects but it is expected to work on similar sort of projects)

1) Projects on formal report writing by documenting an event
2) Letter writing

Course
Syllabus
Semester: II

2.8 VEC (Environmental Awareness)

https://sndt.ac.in/pdf/academics/syllabus-as-per-nep/vec-syllabus/ug- degree/environment-awareness.pdf

Assessment

External Assessment – 50 Marks (as per university syllabus)

Course Syllabus
Semester: II
2.9 Co-Curricular Course
https://www.sndt.ac.in/nep2020/syllabus-as-per-nep/cc-syllabus

Assessment
Internal Assessment – 50 Marks (as per university)