SNDT Women's University, Mumbai

Undergraduate Degree / UG Programme (Syllabus as Per NEP) - Faculty of Science & Technology

Bachelor of Computer Application

(B.C.A.)

As Per NEP – 2020

Semester – I & II

Syllabus
(W.E.F. Academic Year 2025-26)

Terminologies

Vertical	Full-form/Definition	Remarks	Related to Major and Minor Courses
Major (Core)	Subject comprising Mandatory and Elective Courses, Major Specific IKS, Vocational Skill Courses, Internship/ Apprenticeship, Field Projects, Research Projects connected to Major	Minimum 50% of total credits corresponding to Three/Four - year UG Degree- Mandatory Courses	Related to the Major
Minor Course	Course from same or different Faculty	Minimum 18-20 Credits to be completed in the first three years of UG Programme	Related to the Minor
OEC	Open Elective Courses/ Generic courses	10-12 credits to be offered in I and/or II year. Faculty-wise baskets of OEC to be prepared	OEC is to be chosen compulsorily from faculty other than that of the Major
VSC	Vocational Skill Courses, including Hands on Training corresponding to the Major and/or Minor Subject	8-10 credits, to be offered in first three years, wherever applicable vocational courses will include skills based on advanced laboratory practical's of Major	Related to the Majoror Minor
SEC	Skill Enhancement Courses	06 credits, to be offered in I and II year, to be selected from the basket of Skill Courses approved by university	Related to the Major or Minor any relevant Skill
AEC	Ability Enhancement Courses	08 credits, to be offered in I and II year, English: 04 Credits to be earned in Sem - I, Modern Indian Language of 04 credits to be offered in II year	NA
VEC	Value Education Courses	Understanding India, Environmental science/education, Digital andtechnological solutions, Health & Wellness, Yoga education, sports, and fitness	NA

IKS	Indian Knowledge System	Generic IKS Course: basicknowledge of the IKS to be offered at First Year level	Major-Specific IKS Courses: advanced information about the major, part of the major credit to be offered at second- or third-year level
OJT	On-Job Training (Internship/Apprenticeship)	Corresponding to the Major Subject	Related to the Major
FP	Field projects	Corresponding to the MajorSubject	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	NA
CE	Community Engagement andservice		Related to Major
RP	Research Project	corresponding to the MajorSubject	Related to Major

Program Details

Programme Degree		**Bachelor of Computer Application (BCA)**
Parenthesis if any (Specialization)		
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)		After completing this programme, Learner will
	1.	Describe a strong foundation in computer application, including knowledge of Programming languages, Database, 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		A candidate for being eligible for admission to the degree of Bachelor of Computer Applications must have passed the Higher Secondary School Certificate (Std. XII) examination conducted by the different Divisional Boards of the Maharashtra State Board of Secondary and Higher Secondary Education with 45% marks (40% for candidates belonging to Reserved category) with the following subjects :- (from any stream). • English • Any one of the Modern Indian Languages or Modern Foreign Languages or any classical Language or Information Technology/ Any four subjects carrying 100 marks each. **OR** • English • Any one vocational subject carrying 200 marks • Any three subjects carrying 100 marks each. **OR** Must have passed the Higher Secondary School Certificate (Std. XII) examination with the Minimum Competency based vocational courses (MCVC) conducted by the different Divisional Boards of the Maharashtra State Board of Secondary and Higher Secondary Education. • English • Any one of the Modern Indian Languages or Modern Foreign Languages or any classical • Language or Information Technology • General Foundation Course. • Any one subject carrying 300 marks from among the Minimum Competency based vocational courses prescribed by the Higher Secondary School Certificate examination from time to time. **OR** Must have passed an examination of other Board or Body

		Recognized as equivalent thereto. **OR** • Must have passed any three-year Government recognized Diploma programme.
Intake (For SNDTWU Departments and Conducted Colleges)		As per AICTE approval process

Structure with Course Titles

Bachelor of Computer Application

Semester – I

SN	Courses	Type of Course	Credits	Marks	Int	Ext
	Semester I					
10035411	Problem Solving Techniques	Major (Core)	4	100	50	50
10035412	Mathematics Foundations to Computer Science-I	Major (Core)	2	50	50	0
10435411	Digital Marketing	OEC **(Any One)**	4	100	50	50
10435412	Office Automation					
10435413	Introduction to Google Workspace Tools					
10635401	Computer Architecture	VSC S1	2	50	50	0
10735401	Web Technology – I	SEC	2	50	0	50
10810111	English For Academic Writing- Paper I (For Students of English Medium)	AEC (English) **(Any One)**	2	50	0	50
10810112	English Language and Literature- I (For Students of Non-English medium)					
	AEC Link: https://www.sndt.ac.in/pdf/academics/syllabus-as-per-nep/aec-syllabus/ug-degree/ability-enhancement-course.pdf (Available on Website)					
11051112	Indian Knowledge System **IKS Link:** https://www.sndt.ac.in/pdf/academics/syllabus-as-per-nep/iks-syllabus/ug-degree/inception-of-indian-knowledge-system.pdf (Available on Website)	IKS	2	50	0	50

10935401	Introduction to Indian Constitution **Link:** https://www.sndt.ac.in/pdf/academics/syllabus-as-per-nep/vec-syllabus/ug-degree/introduction-to-indian-constitution.pdf (Available on Website)	VEC	2	50	0	50
11450121	Basics of National Service Scheme	CC **(Any One)**	2	50	50	0
11450221	National Cadets Corps. (NCC) Studies- I					
11450322	Health and Wellness					
11450421	Performing Arts Exploration					
	CC Link: https://www.sndt.ac.in/pdf/academics/syllabus-as-per-nep/cc-syllabus/ug-degree/co-curricular-course-as-per-nep-2020-semester-i-syllabus.pdf (Available on Website)					
			22	**550**	**250**	**300**

Semester – II

	Semester II	**Type of Course**	**Credits**	**Marks**	**Int**	**Ext**
20035411	Data Structures	Major (Core)	4	100	50	50
20035412	Mathematics Foundations to Computer Science-II	Major (Core)	2	50	0	50
20335411	Programming with Python	Minor Stream	2	50	50	0
20635411	Operating Systems	VSC S2	2	50	50	0
20435411	Introduction to Open-Source Technology	OEC **(Any One)**	4	100	50	50
20435412	Introduction to Multimedia Applications					
20435413	Fundamentals of Computer					
20735401	Object Oriented Programming using Java	SEC	2	50	50	0
20810111	English For Academic Writing-Paper II (For Students of English Medium)	AEC **(Any One)**	2	50	0	50
20810112	English Language and Literature- II (For Students of Non-English medium)					
	AEC Link: https://www.sndt.ac.in/pdf/academics/syllabus-as-per-nep/aec-syllabus/ug-degree/ability-enhancement-course.pdf (Available on Website)					
20952111	Environment Awareness **Link:** https://www.sndt.ac.in/pdf/academics/syllabus-as-per-nep/vec-syllabus/ug-degree/environment-awareness.pdf (Available on Website)	VEC	2	50	0	50

21450121	Volunteerism and National Service Scheme	CC **(Any One)**	2	50	50	0
21450221	National Cadets Corps. (NCC) Studies- II					
21450323	Yoga Education					
21450421	Fine Art					
	CC Link: https://www.sndt.ac.in/pdf/academics/syllabus-as-per-nep/cc-syllabus/ug-degree/co-curricular-course-as-per-nep-2020-semester-ii-syllabus.pdf (Available on Website)					
			22	**550**	**300**	**250**

Exit with UG Certificate with 4 extra credits (44 + 4 credits)

Course Syllabus

Semester - I

.1.1 Major (Core)

Course Title	**Problem Solving Techniques**
Course Credits	**4 Credits**
	1. Familiarize themselves with basic terminology of computers, demonstrate knowledge of problem-solving concepts, describe programming languages and their evolution.
	2. Create specification from problem requirements by asking questions to is ambiguate the requirement statement.
	3. Design the solution from the specification of a problem and write pseudo code of the algorithm using basic building blocks or structured programming constructs (Sequence, Selection and Repetition statement).
	4. Translate an algorithm into a C computer program
	5. 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, Counter Controlled, 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	**After learning the module, learners will be able to**

Outcomes	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.

Assignments towards Comprehensive Continuous Evaluation

Module 1:

- Create a flowchart and algorithm for a simple problem (e.g., calculating the factorial of a number).
- Write a pseudo-code for the above problem.
- Convert the pseudo-code into a C program.
- Demonstrate the use of basic input/output functions such as printf() and scanf() in the program.
- Include the use of variables, constants, and data types.

Module 2:

- Write a C program to demonstrate the use of different operators (arithmetic, relational, logical, bitwise, etc.).
- Create examples to illustrate the precedence and associativity of operators and evaluate expressions.
- Include conditional statements and loops in the program to show complex expressions and their evaluations.
- 4. Write C programs using different control structures (if, if-else, switch, while, do-while, for loops). Include programs that utilize nested loops and jumping statements (break, continue, go to).

Module 3:

- Apply various problem-solving techniques for following
- Armstrong Number
- Factorial of a number
- Fibonacci Series
- Palindrome number

Module 4:

- Write a program to handle arrays (one-dimensional and two-dimensional) and pass them to functions. Include pointer to array operations.
- Explain practical implementation of Recursion.
- Create programs that use functions to perform various tasks. Include examples of parameter passing (call by value and call by reference).
- Write a program that includes recursion and demonstrates string manipulation using string functions.
- Develop a program that uses pointers for arithmetic operations and demonstrates the concept of null pointers.

Reference Books:

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

Assessment:
Internal Assessment – 50 Marks

Evaluation Scheme:
Depending on the activities mentioned above a project should be developed for 50 marks. The internal assessment, which is a project evaluation, will be done by conducting a project presentation at the College level, where an External Examiner (Industry Expert or Subject Expert) appointed by the College will be evaluating the project depending on evaluation rubrics given below.

The Rubric will have the following Evaluation Parameters:		
Evaluation Parameters	**Description / Evaluation Points**	**Marks**
Code Implementation	• Accuracy in writing and executing the C program, correct logic, syntax	15
Problem Understanding	• Clarity of problem interpretation and correct approach	10
Output Correctness	• Program compiles without errors and gives correct output	10
Code Quality & Structure	• Proper use of indentation, comments, meaningful variable names, modularity	5
Debugging Skills	• Ability to identify and fix logical/syntax errors during execution	5
Lab Record/Journal	• Completeness, correctness, formatting, and timely submission	5

Given below are two sample projects but it is expected to work on similar sort of projects

Project 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: Using C Language, build a project that performs various matrix operations and demonstrates their real-world applications.

External Assessment: (Marks 50)
End Semester examination of 50 marks for 2 hours duration will be conducted.

.1.2 Major (Core)

Course Title	**Mathematics Foundations to Computer Science - I**
Course Credits	**2 Credits**
	1. Familiarize with computer terminology, demonstrate problem-solving skills, and describe programming languages and their evolution.
	2. This course enables the students to use mathematical models and techniques to analyze and understand problems in computer science
	3. 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
Assignments towards Comprehensive Continuous Evaluation	
Worksheet to be prepared by Teachers on each module and given to students for practice. The correction of the worksheets to be done by the teachers and feedback to be given to the class for better improvement in their end semester examination.	
Explain how logic, sets and relations are applied in Computer Science. Provide an example to support your explanation	

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

External Assessment: (Marks 50)
End Semester examination of 50 marks for 2 hours duration will be conducted

.1.3 A. Open Elective Courses/ Generic (OEC)

Course Title	**Digital Marketing**
Course Credits	**4 Credits**
Course Outcomes	**After Completion of this Course, students will be able**
	1. Demonstrate knowledge of digital marketingfundamentals and their application in the modern marketing mix.
	2. Develop skills in creating and managing digital marketing strategies using various tools and platforms.
	3. Implement effective online campaigns through SEO, socialmedia, email marketing, and other digital channels.
	4. Analyze key performance metrics and optimize marketingstrategies 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 MarketingStrategy
	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 Analyse 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
Assignments towards Comprehensive Continuous Evaluation	
Module 1: • Design and execute an SEO audit for a website. • Perform on-page optimization tasks such as checking for broken links, W3 errors, keyword research, optimizing heading tags, titles, and meta descriptions. • Implement necessary changes based on the audit findings, including fixing broken links, optimizing content, and adjusting meta tags. • Set up essential tools like Google Analytics and Google Webmaster Tools for monitoring website performance and tracking SEO metrics. • Create a comprehensive report summarizing the audit findings, implemented changesand recommendations for further optimization	
Module 2: • Choose a website and perform a basic SEO audit using free tools like: • Google Page Speed Insights • Google Analytics, SEMrush • SEO Site Checkup	
Module 3: • Develop a social media marketing campaign for a business. • Create custom graphics and optimize social media profiles for the selected	

business onmajor platforms like Facebook, Twitter, Instagram, etc.

- Plan and execute a paid advertising campaign using platforms like Google AdWords and social media ads.
- Design and send email marketing campaigns, including content creation, templatebuilding, and email list management.
- Monitor and analyze the performance of the campaigns using relevant metrics andanalytics tools.
- Prepare detailed reports showcasing campaign performance, including insights, keymetrics, and recommendations for improvement

Module 4:

- Create a mock Google Ads campaign for a product, service, or brand using the Google Ads interface:
- Choose campaign goal (e.g., website traffic, leads)
- Define audience (location, language, device
- Choose keywords (with match types)
- Write ad copy (headline, description, URL)
- Set budget and bidding strategy

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:
Internal Assessment: (50 marks)

Evaluation Scheme:
Depending on the activities mentioned above a project should be developed for 50 marks. The internal assessment, which is a project evaluation, will be done by conducting a project presentation at the College level, where an External Examiner

(Industry Expert or Subject Expert) appointed by the College will be evaluating the project depending on evaluation rubrics given below.

The Rubric will have the following Evaluation Parameters:		
Evaluation Parameters	**Description / Evaluation Points**	**Marks**
Practical Task Execution	Implementation of lab assignments like SEO, SEM, content creation, campaign setup, etc.	10
Tool Usage Proficiency	Effective use of tools such as Google Analytics, AdWords, SEMrush, Canva, Mailchimp, etc.	10
Creativity and Content Quality	Originality and relevance of ad copies, social media posts, blogs, email marketing, etc.	10
Results & Performance Analysis	Use of KPIs, interpreting metrics like CTR, bounce rate, impressions, engagement, etc.	5
Lab Report/Documentation	Structure, completeness, insights, and submission quality of lab work or project reports	5
Viva Voce	Conceptual understanding of digital marketing strategies and tool functionalities	5
Attendance and Discipline	Regular attendance, punctuality, and active participation in lab sessions	5

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, analyze 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)

External Assessment: (Marks 50)
End Semester examination of 50 marks for 2 hours duration will be conducted

.**1.3 B. Open Elective Courses/ Generic (OEC)**

Course Title	**Office Automation**
Course Credits	**4 Credits**
	1: To Understand the components of office automation
	2: To understand creation of documents using MS Word
	3: To understand creation of worksheet using MS-Excel
	4: To prepare effective presentations using MS-PowerPoint
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	To prepare word document with proper settings
	To learn various file operations
	To learn shortcut keys related to various operations
Content Outline	**INTRODUCTION TO MS WORD**- Working with Documents -Opening & Saving files, Editing text documents, Inserting, Deleting, Cut, Copy, Paste, Undo, Redo, Find, Search, Replace, Formatting page & setting Margins, Using Tool bars, Ruler, Using Icons, using help, Formatting Documents - Setting Font styles, Font selection- style, size, color etc, Type face - Bold, Italic, Underline, Case settings, Highlighting, Special symbols, Setting Paragraph style, Alignments, Indents, Line Space, Margins, Bullets & Numbering. Setting Page style - Formatting Page, Page tab, Margins, Layout settings, Paper tray, Border & Shading, Columns, Header & footer, Setting Footnotes & end notes – Shortcut Keys; Inserting manual page break, Column break and line break, Table of Contents, Index, Page Numbering, date & Time, Author etc., Creating Tables- Table settings, Borders, Alignments, Insertion, deletion, Merging, Splitting, Sorting, and Formula, Drawing - Inserting ClipArts, Pictures/Files etc.,
Module 2 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	To understand creation of worksheet and workbook
	To understand creation of Tables and it’s Properties
	To understand various formulas applying on worksheet

Content Outline	**INTRODUCTION TO MS Excel**- Spreadsheet basics, Screen overview -Titlebar , Ribbon, formatting toolbar(Ribbon), Command group ,Scrollbar, ruler, title bar, statusbar, File tab, Quick access toolbar, Windows Controls, cell, worksheet,workbook,Shortcut keys, file tab-New,Open, Save, Save as.Home Tab-ClipboardCut, copy, paste,Formatpainter.FontEffetcs-fontsize, font(name/face),FontColor, Bold,Italic,Underline,Border, Fillcolor. Paragraph-Alignment, Orientation,WrapText,Merge&Center.Cells-Insert,Delete,Format.InsertTab-Table,Illustrations-Picture,ClipArt,Shapes, SMartArt,Chart,ScreenShot.Charts.Link-Hperlink,Text-Text Box,Header& Footer,WordArt, Object,Symbols-Equation, Symbol.PageLayout-PageSetup-Margins,Orientation,Size,Print Area,PrintPreview,Print.Styles-ConditionalFormatting,Format as Table, Cell Styles. Editing- AutoSum, Clear,Sort&Filter,Find,Replace,Select,GoTo.Formulas-insertFunction-Sum,Average,Min,Max,Count,Today,Date,Day,Month, Year,Len.
Module 3 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	To understand how to create presentations using various slides ,effects and Animations
	To understand shortcut keys
Content Outline	**INTRODUCTION TO MS Powerpoint-**Getting Started in powerpoint,Introduction, Screen Overview,Creatinga Presentation.ShortcutKeys.HomeTab-Clipboard,Creating&editingslides,SlideLayout,FontEffects-Fontsize, name,bold,italic,underline, textcolor,fillcolor,Increasefontsize,Decreasefontsize, Clearformatting,Characterspacing,TextShadow,Strikthrough.Paragraph-Alignment, Columns, Bullets,Numbering,Text Direction,Align Text,ConvertText toSmartArt, Drawing-Shapes,Arrange,Quickstyles,Shapefill, Shapeoutline,Shapeeffects. Editing- Find,Replace,Select.InsertTab-TabIe,Picture,ClipArt,Screenshot,PhotoAlbumShapes,SmartArt,Chart, HyperlinkAction, Text Box, Header and footer, WordArt, Date & Time, Slide,Number, Equation,Symbol,Object,Video,Audio. DesignTabs-Themes,colors,Effects,backgroung,Transitionto slide,Animation, Slide show, presentation
Module 4 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	To undestand use of internet
	To understand creation, sending and receiving email.
Content Outline	**Introduction to Internet:** What is Internet?, Use of Internet, Concept -Site, WWW, Http, Web browser, Web server, search engine, social sites, domain name, server down, online, email or mail, attachment, compose, cc, bcc. Connect to Internet, Email-create email, send email, send email with attachment,

	check received mail, download attachment of received mail

References:

Microsoft Office 2007 Bible - John
Introduction to Information Technology - Alexis Leon, Mathews Leon, and Leena Leon, Vijay Nicole Imprints Pvt. Ltd., 2013.
A Conceptual Guide to OpenOffice
Computer & Internet Basics Step-by-Step - Etc-end the Clutter - Infinity Publishing
Websites: 1) http://windows.microsoft.com/en-in/windows/msoffice-basics-all- topics

Classroom Activities:

Worksheet to be prepared by Teachers on each module and given to students for practice. The correction of the worksheets to be done by the teachers and feedback to be given to the class for better improvement in their end semester examination.

Q: Preapre resume using MS-word
Q:Prepare student mark sheet using MS-Excel
Q: Prepare presentation on "Computer Generation" using MS-PowerPoint

Assessment

Internal Assessment: (50 marks)

Evaluation Scheme:

Depending on the activities mentioned above a project should be developed for 50 marks. The internal assessment, which is a project evaluation, will be done by conducting a project presentation at the College level, where an External Examiner (Industry Expert or Subject Expert) appointed by the College will be evaluating the project depending on evaluation rubrics given below.

The Rubric will have the following Evaluation Parameters:		
Evaluation Parameters	**Description / Evaluation Points**	**Marks**
Conceptual Understanding	Clarity of architecture concepts like pipelining, cache memory, instruction sets, etc.	10
Technical Accuracy	Correct outputs, waveforms, or test cases (if applicable)	15
Practical Application	Correct implementation of concepts in assignments. Use of real-world scenarios.	10
Lab Record/Journal Submission	Completeness, accuracy, and presentation of reports or circuit diagrams	5
Creativity and Design Skills	Innovative layout, design, or interactivity.	5
Timely Submission	Submitted on or before deadline. Late submissions may deduct up to 5 marks unless prior approval.	5

External Assessment: (Marks 50)

End Semester examination of 50 marks for 2 hours duration will be conducted

.1.3 C. Open Elective Courses/ Generic (OEC)

Course Title	**Introduction to Google Workspace Tools**
Course Credits	**4 Credits**
Course Outcomes	**After Completion of this Course, students will be able**
	1. Familiarize the use Google Workspace tools for productivity, collaboration, and communication.
	2. Develop proficiency in document creation, data management, presentations, scheduling, and real-time collaboration.
	3. Encourage cloud-based workflow and digital literacy.
	4. Enhance problem-solving and organizational skills through integrated use of Google tools across various tasks and projects.
	5. Promote responsible digital behavior, including account security, data privacy, and ethical use of collaborative platforms.
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Differentiate from traditional office productivity tools (e.g., Microsoft Office).
	• Navigate cloud-based tools, and understanding basic cloud computing concepts including storage and accessibility.
	• Effectively manage files using Google Drive
	• Create basic Google Formsor surveys or quizzes, including setting question types, collecting responses, and analyzing results**.**
Content Outline	**Introduction to Google Workspace** • Overview of Google Workspace ecosystem • Logging in & understanding cloud computing • Differences between Workspace and traditional Office tools • Account security & data privacy **Google Drive & File Management** • Uploading & organizing files • Sharing permissions and collaboration • File version history and recovery • Using Google Forms for surveys & quizzes
Module 2 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Create and format professional documents using Google Docs
	• Collaborate in real-time on Google Docs and Sheets
	• Perform data entry and analysis in Google Sheets

	• Create and format professional documents using Google Docs**,**
Content Outline	**Google Docs** • Document formatting and layout • Inserting media (images, tables, charts) • Collaboration using comments and suggestions • Using voice typing and add-ons **Google Sheets** • Data entry, formatting & formulas • Charts and conditional formatting • Data validation and filters • Using functions (SUM, IF, VLOOKUP, etc.) • Sharing and protecting sheets
Module 3 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Design and deliver engaging presentations using Google Slides
	• Effectively use Google Meet to schedule, host, and participate in virtual meetings
	• Manage professional communication through Gmail
Content Outline	**Google Slides** • Creating visually appealing presentations • Embedding videos, animations & transitions • Presenter view and audience Q&A • Real-time group presentation editing **Google Meet & Gmail** • Scheduling & conducting meetings • Screen sharing, recording, and captions • Email management: filters, labels, & templates • Integrating Gmail with Calendar & Tasks
Module 4 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Organize and manage time effectively using Google Calendar
	• Utilize Google Keep for efficient note-taking and collaboration
	• Plan and execute a collaborative project by integrating multiple Google Workspace tools
Content Outline	**Google Calendar & Keep** • Event creation & sharing • Reminders, goals, and tasks integration • Using Google Keep for note-taking and collaboration **Collaborative Projects and Integration** • Real-world group project using all tools • Cross-platform integration (Docs, Sheets, Slides in one project) • Case studies & best practices • Presentations and peer evaluations

Assignments towards Comprehensive Continuous Evaluation

Module 1:

- Write a short comparative report (1–2 pages) between Google Workspace and MS Office.
- Upload it to Google Drive, organize it in a folder, and share with view/comment access
- Create a formatted document (e.g., brochure, newsletter) with images, tables, and links.
- Use "Suggesting" mode and comments to review each other's work.
- Add voice-typed content and one add-on (e.g., table of contents or grammar check).

Module 2:

- Create a sheet to track and analyze mock student marks or budget data.
- Apply formatting, formulas (SUM, IF), and at least one chart.
- Use data validation and protect a range.
- Create a 6–8 slide group presentation on any tech or social topic.
- Include transitions, one embedded video, and speaker notes.
- Present using Google Meet with audience Q&A enabled.
-

Module 3:

- Schedule an event in Google Calendar with invitees and a reminder.
- Send a templated email from Gmail inviting attendees.
- Share a checklist or note using Google Keep linked to the event.
-

Module 4:

- Plan and execute a mini project (e.g., event plan, campaign, product launch). Use:

Docs for project plan
Sheets for budget/logistics
Slides for final presentation
Calendar for scheduling
Keep for to-do tracking
Meet for discussion

- Present it to the class and submit all materials via a shared Google Drive folder.

Reference Books:

1. **Maidasani, D. (2021).** *Information and Communication Technology.* Laxmi Publications. Covers computer fundamentals, cloud tools, and basic office productivity apps.
2. **Arora, A. (2020).** *Computer Fundamentals and Applications.* Vikas Publishing House. Chapters on cloud computing, Google Docs, Drive, and online collaboration.
3. **Goyal, A., & Yadav, K. (2022).** *Digital Literacy: Concepts and Applications.* University Science Press. Focuses on digital citizenship, email tools, Google apps, and digital ethics.
4. **Taxali, R. K. (2019).** *Office Automation Tools.* Tata McGraw Hill. Includes MS Office and Google alternatives with comparison.

5. **Madan, S. (2021).** *Computer Applications in Business.* Mayur Publications. Includes practical use of Google Docs, Sheets, Gmail, and internet tools.

Assessment:
Internal Assessment: (50 marks)

Evaluation Scheme:
Depending on the activities mentioned above a project should be developed for 50 marks. The internal assessment, which is a project evaluation, will be done by conducting a project presentation at the College level, where an External Examiner (Industry Expert or Subject Expert) appointed by the College will be evaluating the project depending on evaluation rubrics given below.

Evaluation Parameter	**Description**	**Marks**
Tool Proficiency	Effective and accurate use of Google Workspace tools (Docs, Sheets, Slides, etc.)	10
Creativity & Presentation	Aesthetic quality, formatting, visual design, and originality	10
Collaboration & Sharing	Use of sharing, commenting, permissions, and group participation (when applicable)	10
Task Completion & Relevance	Fulfills assignment objectives with appropriate content and features	10
Integration & Application	Ability to integrate multiple tools for one task (e.g., Meet, Keep, Calendar)	5
Timely Submission & Organization	Submitted before deadline, properly named and organized in folders	5
Total		**50**

Given below are two sample projects but it is expected to work on similar sort of projects.

Project 1: Use Google Workspace tools to collaboratively plan and present a small college fest (music, food, or tech theme)

Project 2: Design and implement your own digital productivity system using Google Workspace

External Assessment: (50 Marks)
End Semester examination of 50 marks for 2 hours duration will be conducted

.1.4 Vocational Skill Courses (VSC)

Course Title	**Computer Architecture**
Course Credits	**2 Credits**
Course Outcomes	**After Completion of this Course, students will be able**
	1. Familiarize with the fundamentals of Digital Electronics and Binary Number Systems
	2.To Learn the implementation of Combinational Circuit
	3.To Learn the implementation of Sequential Circuit
	4.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.**
Activities towards Comprehensive Continuous Evaluation (CCE)	
Module 1 • Number Systems and Computer Arithmetic • Convert the following numbers from one number system to another: Binary to Decimal: 101011 Decimal to Hexadecimal: 175 Octal to Binary: 257 • Perform the following binary arithmetic operations: 1011 + 1101 1001 - 0110 • Explain K-map with the help of an example. • Explain Von Neumann architecture with its significance	
Module 2: • List different types of Flip-Flops. • Describe practical uses of registers in digital electronics. • Explain 4-bit binary counter. • Explain the design and implementation of half adder and full adder with circuit diagram, list	

Reference Books:

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.
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)

Evaluation Scheme:
Depending on the activities mentioned above a project should be developed for 50 marks. The internal assessment, which is a project evaluation, will be done by conducting a project presentation at the College level, where an External Examiner (Industry Expert or Subject Expert) appointed by the College will be evaluating the project depending on evaluation rubrics given below.

The Rubric will have the following Evaluation Parameters:		
Evaluation Parameters	**Description / Evaluation Points**	**Marks**
Conceptual Understanding	Clarity of architecture concepts like pipelining,	10

	cache memory, instruction sets, etc.	
Technical Accuracy	Correct outputs, waveforms, or test cases (if applicable)	15
Practical Application	Correct implementation of concepts in assignments. Use of real-world scenarios.	10
Lab Record/Journal Submission	Completeness, accuracy, and presentation of reports or circuit diagrams	5
Creativity and Design Skills	Innovative layout, design, or interactivity.	5
Timely Submission	Submitted on or before deadline. Late submissions may deduct up to 5 marks unless prior approval.	5

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 Skill Enhancement Courses (SEC)

Course Title	**Web Technology – I**
Course Credits	**2 Credits**
Course Outcomes	**After Completion of this Course, students will be able**
	1. Familiarize with the basics of web technologies and the Internet
	2. Develop skills in designing static and dynamic web pages using HTML and CSS.
	3. Develop proficiency in creating interactive forms and multimedia content for web applications.
	4. 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.

Activities towards Comprehensive Continuous Evaluation (CCE)

Module 1

- Create a static website using HTML, CSS, and JavaScript.
- Design a multi-page website with a consistent layout and navigation menu.
- Use HTML to structure the content of each webpage, including appropriate tags, elements, and attributes.
- Apply CSS to style the website, including selectors, properties, and responsive design techniques for different screen sizes.
- Implement JavaScript to add interactivity to the website, such as dynamic content manipulation and form validation.
- Ensure the website adheres to web standards and best practices for accessibility and Usability

Module 2

- Develop a web application with database integration and secure coding practices.
- Choose a web development framework (e.g., React, Angular, Vue.js) and use it to build a dynamic web application.
- Implement client-side scripting using JavaScript for event handling, form validation, and dynamic content generation.
- Use server-side scripting languages (e.g., Python, Ruby) to handle user requests and interact with a backend database.
- Design and implement database connectivity, including creating SQL queries for data manipulation and integration with the web application.

- Apply secure coding practices to safeguard the web application against common vulnerabilities such as XSS (Cross-Site Scripting) and SQL injection.
- Test the web application thoroughly to ensure functionality, security, and performance.

Reference Books:

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

Assessment:
Internal Assessment: (50 marks)

Evaluation Scheme:
Depending on the activities mentioned above a project should be developed for 50 marks. The internal assessment, which is a project evaluation, will be done by conducting a project presentation at the College level, where an External Examiner (Industry Expert or Subject Expert) appointed by the College will be evaluating the project depending on evaluation rubrics given below.

The Rubric will have the following Evaluation Parameters:		
Evaluation Parameters	**Description / Evaluation Points**	**Marks**
Implementation of Web Pages	Proper use of HTML, CSS, JavaScript for layout, styling, and interactivity	10
Backend/Server-Side Logic	Use of server-side technologies (PHP, Node.js, Python, etc.) and database integration	10
Functionality and Output	Correct working of features (form submission, validations, dynamic content, etc.)	10
UI/UX Design Principles	Aesthetics, layout consistency, responsiveness, accessibility	5
Code Quality and Documentation	Indentation, comments, folder structure, readme files, version control (Git)	5
Viva Voce	Conceptual understanding of HTML, CSS, JS, HTTP, client-server model, etc.	5
Attendance and Conduct	Regularity, discipline, and active participation	5

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 form

Semester - II

.2.1 Major (Core)

Course Title	**Data Structures**
Course Credits	**4 credits**
Course Outcomes	**1.** Demonstrate Data Structures concepts and their practical applications
	2. Apply problem-solving skills using Data Structures.
	3. Analyze Data Structures using C programming language.
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 a t 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.
Assignments towards Comprehensive Continuous Evaluation	

Module 1: Arrays, Searching, and Sorting
• Create and Traverse a Linear Array.
Write a program to create a 1D array and traverse its elements.
• Insert and Delete Elements in an Array.
Write a program to insert and delete an element at a specific position.
• Linear Search and Binary Search
Write separate programs to implement:
  • Linear Search
  • Binary Search (on a sorted array)
  • Sorting Algorithms
• Write programs to sort an array using:
  • Bubble Sort
  • Selection Sort	

- Insertion Sort
- Matrix Operations (2D Arrays)
  - Write a program to perform addition of two 2D matrices.
  - Sparse Matrix Representation
  - Write a program to represent a sparse matrix using a compact array.

Module 2: Linked Lists and Hashing

- Singly Linked List Implementation
  Write a program to create a singly linked list and perform insertion, deletion, and traversal.
- Doubly Linked List
  Write a program to Implement a doubly linked list with insert and delete operations at both ends.
- Circular Linked List
  Write a program to create a circular linked list and perform basic operations.
- Polynomial Addition using Linked List
  Write a program to add two polynomials using a linked list.
- Expression Conversion and Evaluation
  - Convert an infix expression to postfix.
  - Evaluate a postfix expression.
  - Collision Resolution
- Demonstrate collision resolution using:
  - Chaining
  - Open Addressing (Linear Probing)

Module 3: Stacks, Queues, and Recursion

- Stack Using Array and Linked List
  Write a program to implement a stack using arrays.
  Write a program to implement a stack using a linked list.
- Expression Conversion and Evaluation
  Convert an infix expression to postfix.
  Evaluate a postfix expression.
- Recursive Algorithms
  Write recursive programs to compute:
  - Factorial
  - GCD of two numbers
- Differentiate between simple queue, circular queue, and double-ended queue.
- Circular Queue
  Write a program to implement a circular queue using an array.

Module 4: Trees and Graphs

- Binary Tree Traversal
  Implement in-order, pre-order, and post-order traversals.
- Binary Search Tree (BST)
  Create a BST and perform insertion, deletion, and search operations.
- AVL Tree Implementation
  Implement insertion and deletion in an AVL Tree.
- Graph Representation
  Represent a graph using:
  - Adjacency Matrix
  - Adjacency List
- Graph Traversal
  - Implement BFS (Breadth-First Search)
  - Implement DFS (Depth-First Search)
  - Explain in-order, pre-order, and post-order traversals

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

Assessment:
Internal Assessment – 50 Marks

Evaluation Scheme:
Depending on the activities mentioned above a project should be developed for 50 marks. The internal assessment, which is a project evaluation, will be done by conducting a project presentation at the College level, where an External Examiner (Industry Expert or Subject Expert) appointed by the College will be evaluating the project depending on evaluation rubrics given below.

The Rubric will have the following Evaluation Parameters:		
Evaluation Parameters	**Description / Evaluation Points**	**Marks**
Conceptual Understanding	Ability to define, describe, and explain core data structures concepts clearly and accurately.	10
Practical Implementation	Competence in coding data structure operations in C, with focus on correctness, memory handling, and efficiency.	15
Problem Solving & Analysis	Ability to choose appropriate data structures, analyze complexity, and solve given problems effectively.	10
Design & Application	Skill in applying data structures to real-world problems (e.g., graphs, trees, queues), and implementing custom solutions.	5
Timely Submission	Submitted within the deadline. Late submission deducts up to 5 marks unless exempted.	10

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)

External Assessment – 50 Marks
End Semester examination of 50 marks for 2 hours' duration will be conducted

.2.2 Major (Core)

Course Title	**Mathematics Foundations to Computer Science – II**
Course Credits	**2 Credits**
Course Outcomes	1. Develop critical thinking skills to construct and evaluate logical arguments and proofs.
	2. This course introduces mathematical techniques that are foundations for understanding advanced computational methods, including numerical methods and optimization.
	3.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)	
Learning Outcomes	**After learning the module, learners will be able to**
	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)	
Learning Outcomes	**After learning the module, learners will be able to**
	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.
Activities towards Comprehensive Continuous Evaluation (CCE)	
Worksheet to be prepared by Teachers on each module and given to students for practice. The correction of the worksheets to be done by the teachers and feedback to be given to the class for better improvement in their end semester examination.	

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 it's applications*. McGraw Hill Education India.

Assessment:

External Assessment – 50 Marks

End Semester examination of 50 marks for 2 hours duration will be conducted

.2.3 Minor Stream

Course Title	**Programming with Python**
Course Credits	**2 Credits**
Course Outcomes	**1**.Gain proficiency in the basics of Python programming
	2.Develop problem-solving skills using programming.
	3. Acquire skills in programming fundamentals, including variable declaration, looping structures, and function implementation.
	4.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.

Activities towards Comprehensive Continuous Evaluation (CCE)

Module 1: Python Basics & Control Flow

- Hello World & User Input
  Write a program that prints "Hello, Python!" and asks the user for their name and age. Display a greeting message using the input values.
- Simple Calculator
  Create a calculator program that takes two numbers and an operator (+, -, *, /) as input and displays the result.
- Data Types & Type Casting
  Write a program to demonstrate different data types and use int(), float(), and str() for type casting.
- Even or Odd Checker
  Write a program that checks whether a number entered by the user is even or odd using if-else.
- Number Classification
  Take a number from the user and print whether it is positive, negative, or zero using nested if.
- Multiplication Table using Loop
  Write a program to print the multiplication table of a number using a for loop.
- Prime Numbers within a Range
  Write a program to print all prime numbers between two numbers using for and else.

Module 2: Functions, Modular Programming & Error Handling

- Basic Function
  Write a function that takes a number and returns its square. Call it from the main program.
- Function with Default and Keyword Arguments
  Write a function to calculate area of a rectangle. Use default arguments and keyword arguments to demonstrate both.
- Factorial using Recursion
  Create a recursive function to calculate the factorial of a number.
- Lambda Function for Filtering
  Use a lambda function with filter() to get all even numbers from a list.
- Random Number Guessing Game
  Use the random module to generate a number between 1 and 10. Let the user guess until they get it right.
- Exception Handling – Division Program
  Write a program to divide two numbers. Handle Zero Division Error and display a user-friendly message.
- Discuss how can you import and use the math and random modules?

Reference books:

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
End Semester examination of 50 marks for 2 hours duration will be conducted

.2.4 Vocational Skill Courses (VSC S2)

Course Title	**Operating Systems**
Course Credits	2 Credits
Course Outcomes	1.Define the fundamental characteristics and features of operating systems
	2.Comprehend multithreaded programming, CPU scheduling, process management, process synchronization, memory, deadlocks, and storage management.
	3.Compare the performance of CPU scheduling algorithms.
	4. 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.

Activities towards Comprehensive Continuous Evaluation (CCE)

Module 1: Operating Systems Fundamentals & Process Management

- OS Comparison Chart: Create a comparison table for Batch, Time-Sharing, Distributed, and Real-Time OS based on parameters like response time, user interface, and resource handling.
- Structure Drawing: Draw the architecture of a monolithic, layered, and microkernel OS with labeled components.
- Simulation: Use a CPU scheduling simulator to simulate FCFS, SJF, and Round Robin algorithms. Compare their turnaround and waiting times.
- Multithreading Demo: Write a simple multithreading program in Java or Python to show concurrent execution.
- Process State Chart: Design a flowchart to illustrate the life cycle of a process with transitions between states.
- Differentiate between user-level threads and kernel-level threads.
- A process arrives at time 0 with a burst time of 4ms. Show how FCFS, SJF, and RR (quantum 2ms) would schedule it. Calculate turnaround time.
- Group Discussion: Debate the advantages and disadvantages of multiprogramming systems.
- Case Study: Analyze the OS structure of Linux or Windows and present its key features.
- Compare and contrast multiprogramming and time-sharing systems. Define an Operating System. List and explain any four of its components

Module 2: Synchronization, Deadlock, Memory & I/O Management

- Critical Section Role-play: Act out a critical section problem with shared resources and implement a mutual exclusion rule.
- Deadlock Detection Game: Use a resource allocation graph in a group activity to simulate deadlock and recovery strategies.
- Describe how Banker's algorithm prevents deadlocks.
- Memory Allocation Simulator: Simulate memory allocation using MFT, MVT, and show fragmentation.
- Page Replacement Activity: Use charts to demonstrate FIFO, LRU, and Optimal page replacement algorithms.
- Differentiate between semaphores and monitors
- Disk Scheduling Visualization: Create visual demos of SSTF, FCFS, SCAN, and C-SCAN with disk queue examples.
- Virtual Memory Flowchart: Draw the lifecycle of a page fault handling with demand paging.
- Hardware Exploration: Research and present types of I/O hardware and their OS-level management.

Assessment:

External Assessment – 50 Marks
End Semester examination of 50 marks for 2 hours duration will be conducted.

.2.5 A. Open Elective Courses/ Generic (OEC)

Course Title	**Introduction to Open-Source Technology**
Course Credits	**4 Credits**
Course Outcomes	**After Completion of this Course, students will be able**
	6. Understand the concept and philosophy behind open source software.
	7. Identify and use popular open source tools for various tasks.
	8. Install and explore beginner-friendly Linux distributions.
	9. Utilize open source productivity and development tools.
	10. Practice ethical and responsible participation in open source communities.
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Understand the meaning and history of open source.
	• Learn about licenses and community collaboration.
	• Identify advantages and challenges of open source adoption.
Content Outline	Introduction to Open Source - History and evolution - Free Software vs. Open Source - Popular licenses: GPL, MIT, Apache - Benefits and myths Open Source Ecosystem - Communities, contributions, and ethics - Examples: Linux, Firefox
Module 2 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	Install and use a basic Linux OS (Ubuntu/Mint).
	Understand file system structure and command-line basics.
	Use common Linux applications.
Content Outline	Linux for Beginners - What is Linux? Distributions overview - Installing Ubuntu/Linux Mint (live or virtual) - Basic file system navigation - Command line: ls, cd, mkdir, rm, cp, mv - Package management using apt Practical Task: Create folders, install software, edit files
Module 3 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Identify and use open source alternatives to common tools.
	• Use tools for writing, image editing, and communication.

Content Outline	Popular Open Source Software - LibreOffice (Docs, Sheets, Presentations) - GIMP (Image editing) - Thunderbird (Email client) - VLC Media Player Hands-on Practice - Create documents and presentations - Edit an image using GIMP - Install and configure Thunderbird
Module 4 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Understand how to get started with contributing to open source.
	• Plan and execute a basic open source-based mini project.
Content Outline	Getting Involved - Git and GitHub basics - Finding beginner-friendly projects - Reporting issues, documentation, translation Mini Project - Example: Build a simple webpage using open source tools - Document your process - Present findings and reflections
Assignments towards Comprehensive Continuous Evaluation	
Module 1: • Create a digital poster or social media post using Canva.	
Module 2: • Record a short podcast or voice narration and edit using Audacity.	
Module 3: • Create a 1-minute animated explainer video.	
Module 4: • Group project: Develop a 2–3 minute multimedia awareness campaign or tutorial and present it to the class.	

Reference Books

1. Collins, J. F. (2021). *Introduction to Open Source Software*. Open Source Press. Introduces the fundamentals and philosophy of open source development and software distribution.
2. Stallman, R. (2010). *Free Software, Free Society*. GNU Press. A collection of essays exploring the ethics, history, and future of free software by one of its founding figures.
3. Siever, E., Figgins, S., Love, R., & Robbins, A. (2009). *Linux in a Nutshell*. O'Reilly Media. A comprehensive guide to using Linux, including command-line tools, shell scripting, and system administration.
4. Online Help and Community Forums for Ubuntu, LibreOffice, GIMP, and GitHub. Provide practical, community-driven support, documentation, and troubleshooting tips for widely-used open source tools.

Assessment:
Internal Assessment: (50 marks)

Evaluation Scheme:
Depending on the activities mentioned above a project should be developed for 50 marks. The internal assessment, which is a project evaluation, will be done by conducting a project presentation at the College level, where an External Examiner (Industry Expert or Subject Expert) appointed by the College will be evaluating the project depending on evaluation rubrics given below.

Evaluation Parameter	Description	Marks
Tool Proficiency	Effective use of open source tools and Linux environment	10
Creativity & Aesthetics	Design quality and open source tool application	10
Collaboration	Teamwork and responsible contributions	10
Completion & Relevance	Completion of tasks with relevant output	10
Ethics & Community Involvement	Participation in forums, attribution	5
Organization & Timeliness	Timely submission and clear documentation	5
Total		**50**

Given below are two sample projects but it is expected to work on similar sort of projects.

Project 1: Create a Digital Brochure Using LibreOffice and GIMP

To use open-source tools to create a visually appealing brochure about a topic of your choice (e.g., environmental awareness, your college, or open source itself).

Project 2: Simple Website with GitHub Pages and Documentation Contribution

To gain hands-on experience with open-source contribution practices while building a small static website using open-source tools and hosting it on GitHub.

External Assessment: (50 Marks)
End Semester examination of 50 marks for 2 hours duration will be conducted

.2.5 B. Open Elective Courses/ Generic (OEC)

Course Title	**Introduction to Multimedia Applications**
Course Credits	**4 Credits**
Course Outcomes	**After Completion of this Course, students will be able**
	11. Understand the basic concepts and types of multimedia
	12. Use simple tools to create and edit multimedia content (images, audio, video, animations)
	13. Apply multimedia elements to design educational or promotional content.
	14. Enhance digital communication through the use of multimedia tools.
	15. Practice ethical use of multimedia including copyright and digital sharing norms.
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Understand the components and significance of multimedia.
	• Identify types of multimedia and common applications.
	• Use basic image editing tools to enhance visuals.
Content Outline	Introduction to Multimedia - Definition, types (text, image, audio, video, animation) - Multimedia in education, entertainment, and communication Image Editing Basics - Introduction to Photopea, Paint.NET, Canva (Free Tier) - Cropping, filters, resizing, text overlay - Creating posters and banners
Module 2 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Record and edit audio clips using simple software.
	Edit and merge video clips for presentations or social content.
	• Perform data entry and analysis in Google Sheets
	• Create and format professional documents using Google Docs**,**
Content Outline	Audio Editing - Introduction to Audacity, Online Voice Recorder - Trimming, noise removal, effects - Creating voiceovers or podcasts Video Editing - Introduction to Clipchamp / OpenShot - Merging clips, transitions, subtitles - Creating short video messages or reels
Module 3 (Credit 1)	

Learning Outcomes	**After learning the module, learners will be able to**
	• Understand animation concepts and tools.
	• Create basic animations using online tools.
	• Embed interactive media into presentations.
Content Outline	Basics of Animation - Frame-by-frame vs. motion tweening - Using Canva animation tools Interactive Multimedia - Embedding multimedia into Google Slides / PowerPoint - Adding hyperlinks, buttons, and transitions - Creating an interactive story or quiz
Module 4 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	• Plan and execute a multimedia-based mini project.
	• Integrate image, audio, video, and animation in a single product.
Content Outline	Project Planning - Choosing a theme (e.g., awareness campaign, educational tutorial) - Storyboarding and content sourcing Final Integration - Bringing together visuals, audio, video, and animation - Exporting and sharing (YouTube, Drive, etc.) Digital Ethics - Copyright, fair use, attribution - Creative Commons and royalty-free resources
Assignments towards Comprehensive Continuous Evaluation	
Module 1: • Create a digital poster or social media post using Canva.	
Module 2: • Record a short podcast or voice narration and edit using Audacity.	
Module 3: • Create a 1-minute animated explainer video.	
Module 4: • Group project: Develop a 2–3 minute multimedia awareness campaign or tutorial and present it to the class.	

Reference Books:

1. Aughan, T. (2020). *Multimedia: Making It Work*. McGraw Hill. Covers multimedia fundamentals, tools, and applications for interactive content creation.
2. Krishnan, N. (2022). *Multimedia Systems*. PHI Learning. Explains architecture, compression techniques, and synchronization in multimedia systems.
3. Bangia, R. (2019). *Multimedia and Web Technology*. Firewall Media. Focuses on multimedia concepts and integration with web technologies.
4. Tutorials and Help Sections of Canva, Audacity, OpenShot, Animaker. (n.d.). Offers practical guidance on using free tools for graphic design, audio editing, video editing, and animation.

Assessment:
Internal Assessment: (50 marks)
Evaluation Scheme:
Depending on the activities mentioned above a project should be developed for 50 marks. The internal assessment, which is a project evaluation, will be done by conducting a project presentation at the College level, where an External Examiner (Industry Expert or Subject Expert) appointed by the College will be evaluating the project depending on evaluation rubrics given below.

Evaluation Parameter	Description	Marks
Tool Proficiency	Effective use of image, audio, video, and animation tools	10
Creativity & Aesthetics	Design quality and creative content integration	10
Collaboration	Teamwork, sharing, feedback and revision	10
Completion & Relevance	Completion of tasks as per guidelines and project scope	10
Ethics & Resource Use	Proper attribution, copyright, and media sourcing	5
Organization & Timeliness	Submissions as per deadline and well-organized folders	5
Total		**50**

Given below are two sample projects but it is expected to work on similar sort of projects.

Project 1: Educational Infographic Video on "Cyber Safety for Students"

Objective:
To create an engaging and informative multimedia video that educates school or college students on the importance of cyber safety using images, audio narration, and animations.

Project 2: Promotional Campaign for a Local Eco-Club

Objective:
To design a multimedia campaign promoting eco-friendly habits and the work of a local student eco-club.

External Assessment: (50 Marks)
End Semester examination of 50 marks for 2 hours duration will be conducted.

.2.5 C. Open Elective Courses/ Generic (OEC)

Course Title	**Fundamentals of Computer**
Course Credits	**4 Credits**
	1. To Understand basic concepts of computer.
	2. To Understand concept of Computer Hardware.
	3. To Understand concept of Computer Software and operating systems
	4. To Understand windows operating environment
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	Remember and define basic concepts of Computers and their types
	Understand applications of computers in Real life
	Understand history of computers
Content Outline	**Introduction to Computers** : Definition of computer, Characteristics of Computers, Generations of computer, Applications of computer Types of Computers : Digital, Analog, Hybrid Computers, General purpose Computers, Micro Computers, Mini Computers, Mainframes, Super Computers
Module 2 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	Understand working of computer hardware parts
	Understand use of various input, output and storage devices
	Understand types of computer languages
Content Outline	**Introduction to Computer Hardware:** Block diagram of computer-CPU-Input, output **Input Devices**- Keyboard, Mouse, Scanner, smartphone **Output Devices**- Monitor, Printer, Speakers, Headphones **Types of Memory (Primary And Secondary) : RAM, ROM, S**, RAM , ROM, Hard Disk **Types of Computer Programming Languages** : Machine Languages, Assembly Languages and High Level Languages.
Module 3 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	Understand definition, types of software and types of operating systems
	Understand differences between system and Application software
Content Outline	**Introduction to Software & Operating Systems:** Introduction to software, Types of software: System, Application with examples. **Introduction to operating system**: Types of Operating System(Explain concepts): Single processor systems, Multiprogrammed, Batch, Time sharing- Interactive, Multitasking, Multiprocessor systems, Functions of O.S

Module 4 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Understand Windows Operating Environment along with features
	2. Make use of windows accessories
Content Outline	**Windows Operating Environment,** Features of Windows- My Computer, Recycle Bin, Control Panel,Taskbar- start button,start menu,Quick access toolbar, Desktop-Icons, Windows Application, Icons, Windows Accessories : Notepad and MSpaint

References:

1. P. K. Sinha & Priti Sinha , "Computer Fundamentals", BPB Publications, Sixth Edition
2. Silberschatz, Galvin, Gagne "Operating System Principles" John Wiley & Sons, 7th Edition
3. Computer fundamentals by Rajaraman
4. Computer fundamentals, architecture and organisation by B. Ram

Classroom Activities:
Worksheet to be prepared by Teachers on each module and given to students for practice. The correction of the worksheets to be done by the teachers and feedback to be given to the class for better improvement in their end semester examination.

Q: Prepare text file Using notepad and perform file operations like add, update, save data in file.

Q: Draw Nature Scenery using MS-paint

<u>Assessment</u>
Internal Assessment: (50 marks)

Evaluation Scheme:
Depending on the activities mentioned above a project should be developed for 50 marks. The internal assessment, which is a project evaluation, will be done by conducting a project presentation at the College level, where an External Examiner (Industry Expert or Subject Expert) appointed by the College will be evaluating the project depending on evaluation rubrics given below.

The Rubric will have the following Evaluation Parameters:		
Evaluation Parameters	**Description / Evaluation Points**	**Marks**
Conceptual Understanding	Clarity of architecture concepts like pipelining, cache memory, instruction sets, etc.	10
Technical Accuracy	Correct outputs, waveforms, or test cases (if applicable)	15
Practical Application	Correct implementation of concepts in assignments. Use of real-world scenarios.	10
Lab Record/Journal Submission	Completeness, accuracy, and presentation of reports or circuit diagrams	5
Creativity and Design Skills	Innovative layout, design, or interactivity.	5
Timely Submission	Submitted on or before deadline.	5

	Late submissions may deduct up to 5 marks unless prior approval.	

External Assessment: (Marks 50)
End Semester examination of 50 marks for 2 hours duration will be conducted

.2.6 Skill Enhancement Courses (SEC)

Course Title	**Object Oriented Programming using Java**
Course Credits	**2 Credits**
Course Outcomes	1. Apply Object-Oriented Programming Concepts
	2. Demonstrate Proficiency with Java Syntax and Language Fundamentals
	3. Effectively Utilize Arrays, Strings, and Collections
	4. Implement Control Flow and Decision-Making in Java Programs
Module 1 (Credit 1)	
Learning Outcomes	**Learner will be able to:**
	Introduce the object-oriented programming system concepts
	Introduce syntax and semantics of Java programming language
	Develop modular programs using Java
	Setup JDK environment to create, debug and run 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
	Handle packages and use exceptional 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.

Assignments towards Comprehensive Continuous Evaluation

Module 1

- Simple Java Program with Math Functions
  Write a Java program to demonstrate the use of comments and built-in Math functions. Use functions like Math. sqrt(), Math.pow(), and Math.abs() to perform operations on two input numbers.
- OOP Concepts – Class and Object
  Create a class called Employee with attributes like name, ID, and salary. Define methods to set and display employee details.
- Two-Class Interaction
  Design two classes: Calculator (with methods for add, subtract) and MainApp (with main method). Use the Calculator class from MainApp to perform arithmetic operations on user-provided values.
- Command Line Arguments
  Write a Java program that takes a user's name and age as command-line arguments and displays a greeting message including both.
- Variable Declaration, Typecasting, and Constants
  Write a program that declares variables of different data types, performs implicit and explicit typecasting. Display the results of all operations.
- Arithmetic and Relational Operators
  Write a Java program that demonstrates the use of arithmetic and relational operators on two integers. Print results for addition, subtraction, multiplication, division, and comparison (greater, less, equal).
- Logical, Conditional, and Bitwise Operators
  Write a program to demonstrate the use of logical (&&, ||), conditional (ternary), and bitwise (&, |, ^) operators. Use suitable examples to show how these operators work.
- Expression Evaluation and Operator Precedence
  Write a Java program to evaluate a complex expression and demonstrate how operator precedence and associativity affect the result.
- Decision Making with if-else and switch
  Write a Java program that takes an integer input and uses if-else and switch statements to determine whether it is positive, negative, or zero.
- Looping with break, continue, and labeled loops
  Write a program using for loop to print numbers from 1 to 10. Use continue to skip printing number.
  Also demonstrate the use of labeled loop to break out of nested loops when a certain condition is met.
  Compare and contrast the for, while, and do-while loops. When would you use each?

Module 2

- Class and Constructor
  Create a class Book with attributes title, author, and price. Include a constructor to initialize the book details and a method to display them. Create multiple book objects in the main() method and display their details.
- Method Overloading
  Write a Java program with a class Calculator that overloads the add() method to handle:
  - Adding two integers
  - Adding two doubles
  - Adding three integers

  Demonstrate each method from the main() method.
  Method Overriding and Inheritance
- Create a base class Animal with a method make Sound().
- Create subclasses Dog and Cat that override the make Sound() method to print different sounds.
- Call the method from subclass objects and demonstrate runtime polymorphism.
- 1D and 2D Arrays
  Write a Java program to:
  Create a 1D array of integers, input 5 elements, and display the sum and average.
- String Manipulation
  Write a Java program to input a string and perform the following operations:
  Count the number of vowels
  Reverse the string
  Check if the string is a palindrome
- Using Vectors and Wrapper Classes
  Write a Java program that:
  Creates a Vector to store integers using Integer wrapper class
  - Adds 5 numbers to the vector
  - Removes the number at index 2
  - Displays the remaining elements and the size of the vector
- Implementing Interface and Multiple Inheritance
  Create two interfaces Flyable and Swimmable with abstract methods fly() and swim().
  Create a class Duck that implements both interfaces. Show how a class can achieve multiple inheritance in Java using interfaces.
- Creating and Accessing Packages
  Create a package named shapes that includes a class Circle with a method to calculate area. In another file (outside the package), import the shapes package and use the Circle class to display the area for a given radius. Write the steps to create, compile, and run a program using packages.
- Exception Handling with Multiple Catch
  Write a Java program to:
  - Input two numbers and divide them
  - Handle possible exceptions like Arithmetic Exception and Input Mismatch Exception using
    try-catch blocks
  - Include a finally block that prints a closing message
- User Defined Exception
  Create a user-defined exception called Invalid Age Exception. In the main program, take a user's age as input and throw the exception if the age is less than 18. Catch the exception and print a suitable message.

Reference Books:

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

Assessment:

Internal Assessment – 50 Marks

Evaluation Scheme:
Depending on the activities mentioned above a project should be developed for 50 marks. The internal assessment, which is a project evaluation, will be done by conducting a project presentation at the College level, where an External Examiner (Industry Expert or Subject Expert) appointed by the College will be evaluating the project depending on evaluation rubrics given below.

The Rubric will have the following Evaluation Parameters:		
Evaluation Parameters	**Description / Evaluation Points**	**Marks**
Core Functionality & Feature Coverage	Implementation of all core features: OOP principles, control structures, arrays, strings, inheritance, interfaces, exception handling, and packages.	20
Code Structure & Best Practices	Clean, well-structured, modular code following Java naming conventions, and DRY (Don't Repeat Yourself) principles.	10
User Interface & Design Consistency	Intuitive, consistent, user-friendly (even if CLI) UI. Attention to layout, color, and message clarity if GUI is implemented.	10
Input Validation & Exception Handling	Proper validation for user input Exception handling gracefully using try-catch blocks and custom exceptions where appropriate.	5
Project Report & Technical Documentation	Detailed report including objective, methodology, code explanation, screenshots, challenges faced, and conclusion. Documentation of all classes and methods.	5

Given below are two sample projects but it is expected to work on similar sort of projects

Project 1: Using jdk design and implementing 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

References:

1. B.YadavRaju,CMurlikrishna.*AdvantageEnglish*.OrientBlackSwan, 2009.
2. Gangal,J.K.*APracticalCourseinEffectiveEnglishSpeakingSkills.*PHI Learning Private Limited, 2012.
3. Gangal,J.K.*APracticalCourseinDevelopingWritingSkillsinEnglish.* PHI Learning Private Limited, 2011.
4. Swales, John M. and Christine Feak. *Academic Writing for Graduate Students: EssentialTasksandSkills.UniversityofMichiganPressELT*,2012. (3rd Edition)

Assessment
Internal Assessment – 50 Marks

Evaluation Scheme:
Depending on the activities mentioned above a project should be developed for 50 marks. The internal assessment, which is a project evaluation, will be done by conducting a project presentation at the College level, where an External Examiner (Industry Expert or Subject Expert) appointed by the College will be evaluating the project depending on evaluation rubrics given below.

The Rubric will have the following Evaluation Parameters:		
Evaluation Parameters	**Description / Evaluation Points**	**Marks**
Story Analysis (Presentation/Write-up)	Interpretation of characters, plot, and message from stories; language use; clarity of expression	**10**
Grammar & Vocabulary Quiz	Assessment of understanding of tenses, prepositions, subject-verb agreement, and vocabulary	5
Note Taking and Note Making	Ability to extract and organize key points from audio/text; logical and neat presentation	5
Data Interpretation Writing Task	Interpretation of visual data (graphs, tables, charts) and converting into a coherent paragraph/report	5
Formal Letter and Email Writing	Correct format, tone, coherence, and grammar in formal communication (requests or invitations)	10
Speaking Activity (Role Play / Conversation)	Clarity, fluency, grammar, and appropriateness of language in real-life speaking situations	10
Peer Review / Participation	Active class participation and meaningful feedback in peer review exercises	5

Given below are two sample projects but it is expected to work on similar sort of projects

1. Sample Project 1: "Story to Script" – From Narrative to Drama

Project Title: *Adapting a Short Story into a Skit*
Objective:
To develop reading comprehension, interpretation, writing, and speaking skills by transforming a literary story into a short skit for performance.