SNDT Women's University, Mumbai

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

Bachelor of Science (Information Technology)

B.Sc. In IT

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 Major or 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 and technological solutions, Health & Wellness, Yoga education, sports, and fitness	NA
IKS	Indian Knowledge System	Generic IKS Course: basic knowledge 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 Major Subject	Related to the Major
CC	Co-curricular Courses	Health and Wellness, Yoga education sports, and fitness, Cultural Activities, NSS/NCC and Fine/ Applied/Visual/ Performing Arts	NA
CE	Community Engagement and service		Related to Major
RP	Research Project	corresponding to the Major Subject	Related to Major

Program Template

Programme Outcomes (POs) **Action Verbs demonstrating**		After completing this programme, Learner will
	1.	Describe a strong foundation in computer application, including knowledge of Programming languages, Database, Mathematics, Operating system and Networking.
(Major) discipline-related knowledge acquisition, **mastery over cognitive and** **professional, vocational skills are to be used e.g. demonstrate sound understanding of, analyses, compare, create, design, etc...(minimum5)**	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		B. Sc I Year: A candidate for being eligible for admission to the three year course leading to the degree of Bachelor of Science 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 the following subjects : - English Any one of the Modern Indian Languages or Modern Foreign Languages or any classical Language or Information Technology Any four Science subjects carrying 100 marks each. OR English Any one vocational subject carrying 200 marks Any three Science subjects carrying 100 marks each OR Must have passed an examination of any other recognised Board or Body Recognized as equivalent . Students who have not done Mathematics at 12 th Std. are needed to take a bridge course in Mathematics and pass a university conducted test before Semester I examination.

Eligibility Criteria for Programme		B.Sc (IT) II Year: For a student from our University - should have cleared or has ATKT as admissible in first year in the same subject. For a student from another university - studied at least two courses of four credits each in the I year in the same subject. Passed I year without ATKT. Passed three year full time Diploma in Engineering with an aggregate of not less than 60% (Open Category) and 55% (Reserved Category) OR Must have passed any three year Diploma in Engineering Course. B.Sc (IT) III Year: For a student from our University- should have cleared second year in the same subject or has passed with admissible ATKT. For a student from another university– should have completed at least six courses of 4 credits each or 24 credits in the I and II year in the same subject. Passed in I and II year without ATKT.
Intake (For SNDTWU Departments and Conducted Colleges)		As Per University Affiliation Letter

Structure with Course Titles

B. Sc. In Information Technology

Semester - I

SN	Courses	Type of Course	Credits	Marks	Int	Ext
	Semester I					
10135211	C Programming	Major (Core)	4	100	50	50
10135212	Computer Fundamentals and Operating System	Major (Core)	2	50	50	0
10435211	Introduction to Web Technology	OEC **(Any One)**	4	100	50	50
10435212	Fundamentals of Statistics using Spreadsheet					
10435213	Introduction to Cyber Security					
10635201	Office Automation Tools	VSC S1	2	50	50	0
	Swayam / Chetana / MOOC	SEC **(Any One)**	2	50	0	50
10735211	Web Technology - I					
10735212	UI / UXI design					
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)					

11051111	Inception of India 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 (Generic)	2	50	0	50
10952111	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

SN	Courses	Type of Course	Credits	Marks	Int	Ext
	Semester II					
20135211	Data Structure & Algorithms	Major (Core)	4	100	50	50
20135212	Computer Organization and Architecture	Major (Core)	2	50	0	50
20335211	Object Oriented Programming using C++	Minor Stream	2	50	50	0
20635211	Linux Operating system	VSC S2	2	50	50	0
20435211	Multimedia System	OEC **(Any One)**	4	100	50	50
20435212	Introduction to Artificial Intelligence					
20435213	Cyber forensic					
20735201	Vedic Mathematics	SEC **(Any One)**	2	50	50	0
20735202	Mathematics - I					
20810111	English For Academic Writing - Paper II (For Students of English Medium)	AEC (English) **(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:**	VEC	2	50	0	50

	https://www.sndt.ac.in/pdf/academics/syllabus-as-per-nep/vec-syllabus/ug-degree/environment-awareness.pdf (Available on Website)					
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**

Course Syllabus

Semester - I

.1.1 Major (Core)

Course Title	**C Programming**
Course Credits	**4 Credits**
Course Outcomes	1 Apply logic to create programs in C.
	2. Analyze and understand computer programming language concepts.
	3.Evaluate and interpret the use of pointers, including their declarations, initialization, and operations.
	4.Translate an algorithm into a C computer program
	5. Design and develop applications using basic programming constructs, facilitating the transition to other languages.
	6.Apply algorithms by writing C code.
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Learn steps in problem solving using C.
	2.Understand structure, keywords, operators, and functions of C programming.
	3. Evaluate the concept of I/O functions, header files, and preprocessor directives.
	4. Design and apply concepts of the C language.

<table>
<tr><td>Content Outline</td><td>Introduction to problem solving:
Concept: Steps in problem solving - (Define Problem, Analyze Problem, Explore Solution), Problem solving techniques - (Trial & Error, Brainstorming, Divide & Conquer), Algorithms and Flowcharts (Definitions, Characteristics, Advantage & Disadvantages, Symbols, Examples), Pseudo-code (Definition, Conditional statements, Loops),etc.

Overview of programming languages:
Definition of the program, Concept- Source code, Object code, Compilation, Interpretation, Execution, Input and Output, Debugging etc., Expressions, control structures; sub routines, Storage management; scoping rules; bindings for names, Storage types: Automatic, external, register and static variables

Introduction to ‘C’ Language:
History of C Programming, Structures of „C“ , Programming, Simple example, Basic Input/ Output, Function as building blocks. Language Fundamentals:Character Set, C Tokens,Keywords, Identifiers,Variables, Constant, Data Types, Comments

Operators:
Types of operators, Precedence and Associativity, Expression. Statement and types of statements, Built-in Operators and functions. Console based I/O and related built in I/O Function: printf(), scanf(), getch(), getchar(), putchar(), etc; Concept of header files, Preprocess or directives: #include,#define, Conditional statements and Loops</td></tr>
<tr><td colspan="2">Module 2 (Credit 1)</td></tr>
<tr><td rowspan="5">Learning Outcomes</td><td>After learning the module, learners will be able to</td></tr>
<tr><td>1. Gain proficiency in writing C programs to solve various problems.</td></tr>
<tr><td>2. Learn the syntax and semantics of the C language, including its specific features such as pointers and memory management.</td></tr>
<tr><td>3. Analyze the difference between structure and union.</td></tr>
<tr><td>4. Design and handle operations of the files.</td></tr>
</table>

Content Outline	**Control structures** Decision making structures: If, If-else, Nested If –else, Switch, Loop Control structures ,While, Do-while, For, Nested for, while, do-while loop, jumping statements: break, continue, go to, exit. **Functions:** Definition, Basic types of function, Declaration and definition, Function call, Types of function, Parameter passing, Call by value, Call by reference, Scope of variables, Recursion, String: Declaration, string Functions, String Manipulations **Pointers:** Introduction to pointers, Pointer notation, Pointer arithmetic, Null Pointer **Arrays:** Definition, Declaration, Initialization, Bounds checking, One-Dimensional Array, Two-Dimensional Array, Passing array to a function, pointer to Array **Structure and Union:** Introduction to Structure, Definition, Declaration of Structure Variables, Dot Operator, Nested Structure, Array of Structure, pointer to structure, Introduction to Union, Difference between Structure and Union **File Handling:** Concept of File, Definition, File operations (create, open, read, move, write, close), File opening Mode, Closing a file, Input / output operations, Creating and reading a file, Command Line Argument
Module 3 (Credit 1)	**After learning the module, learners will be able to**
Learning Outcomes	1. Apply operators to write simple programs.
	2. Analyze and use control, iterative, and jumping statements.
	3. Evaluate the use of break and continue statements.
	4. Design programs with header files and preprocessor directives.
Content Outline	**Simple Program Implementation of Operators:** Built in Operators and function, Arithmetic, Logical, Relational, bitwise, Precedence And Associativity, composite statements. Unary, binary and ternary operators. **Concept of header files**, Preprocessor directives: #include, #define. And macros implementations, Implementation of Storage types: Automatic external, register and static variables **Console based I/O and related built in I/O function**: printf(), scanf(), getch(),getchar(), putchar();

Content Outline	**Control Statement:** Decision Making Statements, if, Nested if, if-else, Nested if-else, if-else-if, switch, etc. The Conditional Expression; **Iterative Statements**- The for loop, . The while loop, The do-while loop; Jumping Statements- The goto & label, The break & continue, The exit()function.
Module 4 (Credit 1)	**After learning the module, learners will be able to**
Learning Outcomes	1. Apply functions in programs.
	2. Analyze the declaration, initialization of pointers, and passing arrays to functions.
	3. Evaluate the definitions and declarations of structure variables in programs.
	4. Design programs effectively using functions, pointers, and structures.
Content Outline	**Implementation of Functions:** Defining and accessing, passing arguments, Function prototypes, function calling mechanism, call by value, call by reference, recursive function; String Manipulations Pointer Declaration and Initialization of Pointer variables, pointer Arithmetic, Pointers and Character Strings Implementation of 1-D and multidimensional Array, One Dimensional Array, Two-Dimensional Array, Passing array to a function, pointer to Array. **Programs Using Structure and Union:** Defining and Declaring Structure Variables, .Dot Operator, Nested Structure, Array of Structure, pointer to structure, Examples of Union. **Programs using I/O Operations File Handling:** File Operations (Create, open, read, move, write, close) Input/output operations on file Character by –(fgetc, fputc), Reading and writing files
Activities to be done in the class towards Comprehensive Continuous Evaluation	
Module 1: 1. Create a flowchart and algorithm for a simple problem (e.g., calculating the factorial of a number). 2. Write a pseudo-code for the above problem. 3. Convert the pseudo-code into a C program. 4. Demonstrate the use of basic input/output functions such as printf() and scanf() in the program. 5. Include the use of variables, constants, and data types. 6. Write a C program to demonstrate the use of different operators (arithmetic, relational, logical, bitwise, etc.). 7. Create examples to illustrate the precedence and associativity of operators and evaluate expressions. 8. Include conditional statements and loops in the program to show complex expressions and their evaluations. 9. Demonstrate the use of console-based I/O functions such as printf(), scanf(), getch(), getchar(), putchar(), etc. 10. Illustrate the use of header files and preprocessor directives (#include, #define).	

Module 2:

1. 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).
2. Create programs that use functions to perform various tasks. Include examples of parameter passing (call by value and call by reference).
3. Write a program that includes recursion and demonstrates string manipulation using string functions.
4. Develop a program that uses pointers for arithmetic operations and demonstrates the concept of null pointers.
5. Write a program to handle arrays (one-dimensional and two-dimensional) and pass them to functions. Include pointer to array operations.
6. Write programs to perform basic file operations (create, open, read, write, close). Include programs that demonstrate reading from and writing to files.
7. Implement a program that uses command-line arguments for file operations. 8. Compare and contrast the use of structures and unions in handling data through a practical example in a program.

Module 3

Write a C program that implements a simple calculator. The program should prompt the user to enter two numbers and an operator (+, -, *, /). Based on the operator entered, perform the corresponding arithmetic operation and display the result.

1. Use appropriate control flow statements (if, else if, else or switch) to determine the operation to be performed based on the operator entered by the user.
2. Implement error handling to handle division by zero.
3. Utilize console-based input/output functions like printf(), scanf() for user interaction.

Make use of header files and preprocessor directives to organize your code and define any
necessary constants

Module 4

Write a C program that demonstrates the use of functions, pointers, and structures to manage student records. Each student record should contain the following information:
name, roll number, marks in three subjects.

1. Define a structure to represent a student record with appropriate data members.
2. Implement functions to perform the following tasks:
3. Input student details (name, roll number, marks).
4. Calculate the total marks and average marks of a student.
5. Display student details along with total and average marks.
6. Use pointers to pass structures to functions wherever necessary.
7. Ensure proper memory allocation and deallocation.
8. Implement file handling operations to read and write student records to a file using input/output functions like fscanf(), fprintf(), fopen(), fclose(), etc.

Reference Books

1. Schildt, H. (2000). C: The Complete Reference (4th ed.). Tata McGraw-Hill Education Pvt.Ltd.
2. Ramkumar, & Agrawal. (1996). Programming in ANSI C. Tata McGraw-Hill. 3. Kanetkar, Y. P. (2008). Let Us C. Infinity Science Press.
3. The C Programming Language, Authors**:** Brian W. Kernighan & Dennis M. Ritchie
4. C: The Complete Reference, Author**:** Herbert Schildt
5. The C Programming Language, Kernighan & Ritchie

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 creates 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 by the college

.1.2 Major (Core)

Course Title	**Computer Fundamentals And Operating System**
Course Credits	**2 Credits**
Course Outcome	1. Apply knowledge to set up and connect various computer peripherals and interfaces.
	2.Analyze and perform conversions between different number systems and execute binary arithmetic operations.
	3.Basic networking concepts and Evaluate and troubleshoot protocols
	4.Design and optimize operating system components to enhance their functions and overall system performance.
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1.Apply concepts of number systems and computer arithmetic to solve computational problems.
	2.Analyze the differences and functionalities of system and application software, and various types of computers.
	3.Evaluate the characteristics and interpretations of data, including memory devices and computer languages.
	4.Design efficient process scheduling and synchronization mechanisms within various types of operating systems.
Content Outline	**Number Systems**: Binary, Octal, Decimal, Hexadecimal and Their inter conversion, Computer Arithmetic. **Computer Software:** System and Application Software. **Type of Computers:** Digital, Analog, Hybrid Computers **Definition:** Data, Information; Characteristics and Interpretation, Data & it's logical & physical Concepts, Definition of Computer, Features, Block Diagram of Computer System, Computer Generations **Primary Memory Devices:** RAM, ROM, PROM,EPROM, CACHE Memory, Registers. **Computer Languages:** Machine, Assembly, High Level **Operating System:** Purpose of Operating Systems, OS Structure, Services of Operating System,Types of Operating System **Processes**: Concept, process states, Scheduling, Operations on Processes, Cooperating Process, Process Synchronization.

<table>
<tr><td colspan="2">Module 2 (Credit 1)</td></tr>
<tr><td rowspan="5">Learning Outcomes</td><td>After learning the module, learners will be able to</td></tr>
<tr><td>1. Apply CPU scheduling algorithms (FCFS, SJF, RR, Priority) to optimize process scheduling.</td></tr>
<tr><td>2. Analyze various memory management techniques, including swapping,
contiguous memory allocation, paging, and segmentation.</td></tr>
<tr><td>3.Evaluate different page replacement policies (LRU, OPT, SC, FIFO, NRU, MRU) to enhance memory management efficiency.</td></tr>
<tr><td>4. Design memory management strategies incorporating advanced scheduling and page replacement algorithms to improve overall system performance.</td></tr>
<tr><td>Content Outline</td><td>CPU Scheduling: Concept, Scheduling Criteria, Scheduling Algorithms (FCFS, SJF, RR, Priority).
Memory Management: Concept, Swapping, Contiguous Memory Allocation, Paging, Segmentation.
Page Replacement policies: Least Recently used(LRU) Optimal (OPT), Second Chance (SC), First in First Out (FIFO), Not recently used (NRU), Most Recently Used(MRU).</td></tr>
<tr><td colspan="2">Activities to be done in the class towards Comprehensive Continuous Evaluation</td></tr>
</table>

Module 1

Comprehensive Computer Systems Analysis and Implementation 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

Computer Software

1. Compare and contrast system software and application software by listing at least 3 examples of each.
2. Explain the primary functions of each type of software.

Types of Computers

3. Describe the characteristics and uses of digital, analog, and hybrid computers. Provide one real-world example for each type.

Data and Computer Systems

• Define the terms data and information. Explain the difference between the two with examples.

• Draw a block diagram of a computer system and label its main components. • Describe the features and characteristics of each generation of computers from the first to the current generation.

Primary Memory Devices

Explain the differences between RAM, ROM, PROM, EPROM, and CACHE memory. Include their primary uses and characteristics.

Computer Languages

• Differentiate between machine language, assembly language, and high-level languages. Provide one example for each and explain their typical uses in programming.

Operating Systems

• Describe the primary purpose and services provided by an operating system.

• Explain the structure of an operating system with a diagram.

• List and briefly describe different types of operating systems (e.g., batch, time- sharing, distributed).

Processes and Scheduling

• Explain the concept of a process and its states. Illustrate the process state diagram.

• Describe different CPU scheduling algorithms (FCFS, SJF, RR, Priority). Provide a scenario where each algorithm would be most appropriate.

• Explain process synchronization and provide an example where it is crucial in an operating system.

Module 2
To apply, analyze, evaluate, and design CPU scheduling algorithms, memory management techniques, and page replacement policies through a comprehensive simulation and analysis

activity. CPU

Scheduling:

Define and explain the concepts of CPU scheduling and scheduling criteria.

Implement and compare the following scheduling algorithms: FCFS, SJF, RR, and Priority.

• Task: Simulate a scenario with a set of processes, each having different burst times, arrival times, and priorities. Evaluate the performance of each scheduling algorithm based on average waiting time, turnaround time, and CPU utilization.

• Deliverable: Provide a detailed report with your simulation code, a comparison of the results, and an analysis of which algorithm performs best under different conditions.

Memory Management:

Explain the concepts of memory management, including swapping, contiguous memory allocation, paging, and segmentation.

• Task: Simulate a memory management system that includes swapping and both contiguous and non-contiguous allocation methods (paging and segmentation).

• Deliverable: Create a memory map for each allocation method. Provide your simulation code, memory maps, and an analysis of the advantages and disadvantages of each method.

Page Replacement Policies:

Define and explain the following page replacement policies: LRU, OPT, SC, FIFO, NRU, and MRU.

• Task: Simulate a scenario where a sequence of page requestsis given. Implement each of the page replacement policies and evaluate their performance based on the number of page faults.

• Deliverable: Provide your simulation code, a comparison of the number of page faults for each policy, and an analysis of which policy is most effective under different conditions.

References book:

1. Stallings, W. (2021). Computer organization and architecture: Designing for performance (11th ed.). Pearson Education.
2. Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). Operating system concepts (10th ed.). Wiley.
3. Tanenbaum, A. S., & Austin, T. (2012). Structured computer organization (6th ed.). Pearson Education.
4. Patterson, D. A., & Hennessy, J. L. (2017). *Computer organization and design: The hardware/software interface* (5th ed.). Morgan Kaufmann.
5. Stallings, W. (2018). *Operating systems: Internals and design principles* (9th ed.). Pearson 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**
Understanding of Computer Components	Identify and explain components of a computer system (CPU, RAM, storage devices, I/O devices). - Differentiate between system and application software.	10
Operating System Concepts	- Describe core OS functions: process management, memory management, file systems, and device management. - Explain the role of the OS in resource allocation.	10
Practical OS Skills	- Demonstrate proficiency in basic OS operations: booting, user account management, file navigation. - Perform tasks like installing/uninstalling software and configuring system settings.	15
Problem-Solving & Troubleshooting	Analyze and resolve common OS issues (e.g., application crashes, slow performance). - Utilize system tools for diagnostics and maintenance.	10
Communication & Presentation	Effectively communicate technical information. - Present findings and solutions clearly, using appropriate terminology.	5

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

Project 1: Identification of Computer Components:

List and briefly explain the functions of key computer parts like CPU, RAM, Storage devices, Keyboard, and Monitor.
Difference Between System Software and Application Software,Provide two examples each of system software and application software with a short explanation of each.

Project 2: Operating System Internals:

Study and explain core OS functions:

Process Scheduling Algorithms (e.g., Round Robin, Priority Scheduling)
Memory Management techniques (paging, segmentation, virtual memory)
File System architecture and permissions
Device management and I/O handling
Illustrate these with real-world examples or simulations.

.

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

Course Title	**Introduction to WEB TECHNOLOGY (HTML, CSS, PHP, JAVASCRIPT)**
Course Credits	**4 Credits**
Course Outcomes	1. Apply HTML5 and CSS to create and style web pages effectively.
	2. Analyze and implement JavaScript to develop interactive web pages
	3. Evaluate and develop dynamic, database-driven web pages using PHP.
	4. Design and implement comprehensive client-side and server-side scripting programs.
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Apply HTTP concepts and security measures to manage client server interactions, including persistent connections and cookie
	2. Analyze and utilize HTML5 elements and tags to create structured and semantic web pages.
	3. Evaluate the effectiveness of HTML5 web forms in capturing various types of user input, ensuring usability and accessibility
	4. Design and implement multimedia content and interactive form elements in web pages to enhance user experience.
Content Outline	**Introduction to web and Security Concepts** HTTP: Overview – HTTP Basics, Client request, Server response; HTTP Headers; Session Management – Persistent connections, Cookies. General concepts on web server: virtual hosting, General concepts of caching proxy server, Web security, Digital signatures, Digital Certificates, Encryption, and Authentication **HTML5** Basics of HTML elements and Tags. Introduction of HTML5 (evolutions, limitation of HTML4, advantages of HTML5, Overview of HTML5) **Page Layout of Semantic Elements** Header,Navigation, Section & Articles, Footer Organizing Text in HTML, Links and URLs in HTML, Tables in HTML, Images on a Web Page, Image Formats, Image Maps, Colors, FORMs in HTML, Frames in HTML Working with Multimedia -Inserting Audio and Video on a web page,audio video file format. **HTML5 Web Forms** HTML 5 Global Attributes Displaying a Search Input

	Field, Contact Information Input Fields, Utilizing Date and Time Input Fields, Number Inputs, selecting from a Range of Numbers, Selecting Colors, Creating an Editable Drop-Down, requiring a Form Field, autofocusing a Form Field, Displaying Placeholder Text.
Module 2 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. CSS syntax, selectors, and properties effectively to style web pages, including background settings, fonts, text styles, and element positioning
	2. Analyze and implement JavaScript programming fundamentals such as variables, operators, control flow statements, and core JavaScript objects (Array, Boolean, Date, Function, etc.), along with handling events and using browser objects.
	3. Evaluate the integration of PHP with HTML for server-side scripting, understanding syntax, variables, and passing information between web pages.
	4. Design and implement advanced JavaScript functionalities including the Document Object Model (DOM) manipulation, form validation techniques, and utilizing cookies for enhanced web application interactivity and functionality.
Content Outline	**CSS:** Understanding the Syntax of CSS, CSS Selectors, Inserting CSS in an HTML Document, CSS properties to work with- background of a Page, CSS Fonts and Text Styles, positioning an element **JavaScript:** Using JavaScript in an HTML Document, Programming Fundamentals of JavaScript - Variables, Operators, Control Flow Statements, Popup Boxes, Core JavaScript (Properties and Methods of Each) : Array, Boolean, Date, Function, Math, Number, Object, String, regExp, Events and Event Handlers, Browser Objects - Window, Navigator, History, Location, Document, Cookies, Document Object Model, Form Validation using JavaScript. **INTRODUCTION TO PHP AND SQL**: - Server-side web scripting, Installing PHP, SQL, Adding PHP to HTML, Syntax and Variables, Passing information between pages.
Module 3(Credit 1)	

Learning Outcomes	**After learning the module, learners will be able to**
	1. Apply basic HTML tags to design web pages with organized text, links, tables, images, forms, and multimedia elements
	2. Analyze and utilize HTML5 semantic elements (navigation, section, articles, footer, etc.) to structure web page content effectively.
	3. Evaluate and implement CSS syntax and properties to enhance web page aesthetics, including backgrounds, fonts, text styles, and element positioning
	4. Design interactive and multimedia-rich web pages by integrating HTML and CSS, ensuring proper use of image maps, audio, and video file formats
Content Outline	**Use of Basic Tags**, **Image maps, Tables, Forms and Media Design** webpages using the given tools in HTML
	Navigation, Section & Articles, Footer, aside and more. Organizing Text in HTML, Links and URLs in HTML, Tables in HTML, Images on a Web Page, Image Formats, Image Maps, Colors, FORMs in HTML, Frames in HTML Interactive Elements, Working with Multimedia - Audio and Video File Formats, HTML elements for inserting Audio / Video on a web page • **CSS** Syntax of CSS, CSS Selectors, Inserting CSS in an HTML Document, CSS properties to work with background of a Page, CSS properties to work with Fonts and Text Styles, CSS properties for positioning an element
Module 4(Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Apply control flow and looping statements in JavaScript to create dynamic and interactive web pages
	2. Analyze and utilize core JavaScript properties and methods (Array, Boolean, Date, Function, Math, Number, Object, String, RegExp) along with events and event handlers to enhance web page functionality
	3. Evaluate and implement PHP in web development, including syntax, variables, and the integration of PHP with HTML for passing information between pages.
	4. Design and develop database-driven web applications using PHP and SQL, demonstrating the installation and configuration of PHP and the creation of dynamic content

Content Outline	**Java Script :** Control and looping statements and Java Script reference, Using JavaScript design, a web page; Control Flow, Statements, Design a web page demonstrating different conditional statements. Design a web page demonstrating different looping statements; Popup Boxes, Core JavaScript (Properties and Methods of Each) : Array, Boolean, Date, Function, Math,Number, Object, String, regExp, Events and Event Handlers. **PHP & SQL** Demonstrate program in PHP, Installing PHP, SQL, Adding PHP to HTML, Syntax and Variables, Passing information between pages

Activities to be done in the class towards Comprehensive Continuous Evaluation

Module 1:

1. Web and Security Concepts:

• Explain the basics of HTTP, including client requests and server responses. Illustrate with an example of HTTP headers used in a web request.

• Describe session management techniques such as persistent connections and cookies. Create a simple HTML page demonstrating the use of cookies.

2. HTML5 Basics:

• Write an HTML5 document that includes various HTML elements and tags. Highlight the advantages of HTML5 over HTML4.

• Create a simple web page layout using semantic elements such as header, navigation, section, articles, and footer.

3. Page Layout of Semantic Elements:

• Develop a web page that organizes text, includes links, and embeds tables and images. Use different image formats and demonstrate the use of image maps.

• Create a form in HTML5 with various input fields, including search, contact information, date and time, number inputs, and color picker. Include attributes like autofocus, placeholder text, and required fields.

4. Working with Multimedia:

• Add audio and video elements to a web page. Provide examples of different audio and video file formats that can be used.

Module 2:

1. CSS:

• Create an HTML document and apply various CSS styles. Use different selectors and properties to style the background, fonts, text, and position elements.

• Develop a responsive web page layout using CSS for positioning elements.

2. JavaScript:

• Write JavaScript code to demonstrate the use of variables, operators, and control flow statements. Include examples with popup boxes.

• Create a web page with JavaScript that manipulates core objects like Array, Boolean, Date, Function, Math, Number, Object, String, and RegExp. Include event handlers and form validation.

3. Introduction to PHP and SQL:

• Install PHP and configure it with a web server. Write a PHP script that integrates with an HTML form to capture user input.

• Develop a simple web application that uses PHP to interact with a SQL database. Implement functionality to add, retrieve, update, and delete records.

Module 3:

Create a comprehensive web page incorporating basic HTML tags, image maps, tables, forms, and multimedia elements, and style it using CSS.

1. HTML Structure:

Design a web page using HTML that includes a header, navigation bar, sections, articles, aside, and footer.
Organize text with proper HTML tags.
Insert links and URLs.
Create tables to display data.
Add images in various formats and create image maps.
Develop forms to capture user input with various input fields. Include interactive elements using frames
Embed audio and video files using HTML multimedia elements.

2.CSS Styling:

Apply CSS to style the web page.
Use CSS selectors and properties to customize the background, fonts, text styles, and positioning of elements.
Ensure the web page is visually appealing and follows a consistent design theme.

Module 4:

Develop a dynamic web application using JavaScript for client-side scripting and PHP with SQL for server-side scripting

1. JavaScript Functionality:

• Design a web page with JavaScript to demonstrate different control flow statements (if else, switch) and looping statements (for, while).

• Implement popup boxes and core JavaScript objects (Array, Boolean, Date, Function, Math, Number, Object, String, RegExp).

• Add event handlers to manage user interactions and form validations.

2.PHP and SQL Integration:

• Install PHP and configure it with your web server.

• Write PHP scripts to process form data and interact with a SQL database. • Create a database and implement CRUD (Create, Read, Update, Delete) operations. • Pass information between web pages using PHP.

Reference Books: -

1. Powell, T. (2000). Web design: The complete reference. Tata McGraw-Hill.
2. Powell, T. (2008). HTML and XHTML: The complete reference. Tata McGraw-Hill.
3. Powell, T., & Schneider, F. (2004). JavaScript 2.0: The complete reference (2nd ed.). Tata McGraw-Hill.
4. Holzner, S. (2017). PHP: The complete reference. Tata McGraw-Hill.
5. Lecky-Thompson, G. W. (2009). Web programming. Cengage Learning.

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

HTML/CSS design a single-page resume builder using HTML forms.

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	**Fundamentals of Statistics using Spreadsheets**
Course Credits	4 Credits
Course Outcomes	1. Understand the basic concepts and applications of statistics.
	0. Use spreadsheets effectively for data handling and statistical analysis.
	0. Visualize data and interpret results using appropriate charts and graphs.
	0. Perform descriptive and inferential statistical analyses.
	0. Develop data-based reasoning and critical thinking.
Module 1 (Credit 1)	
Learning Outcomes	After learning the module, learners will be able to
	1. Explain the significance and application of data in real-world contexts
	2. Differentiate between qualitative and quantitative data. Identify appropriate methods of data collection and assess their merits and limitations.
	3. Classify data based on types (primary vs secondary) and levels of measurement.
	4. Use basic spreadsheet tools to enter, format, and organize raw data.

Content Outline	**Introduction to Statistics and Spreadsheets** • Types of data: qualitative vs quantitative • Levels of measurement • Introduction to Excel/Google Sheets interface • Basic spreadsheet operations and formulas. **Introduction to different types of data** • Relevance of data in the current scenario • Types of data : primary and secondary data • Methods of data collection: Questionnaire method, Interview method, case study method in brief • Sources of secondary data collection with merits and Limitations. • Classification of data
Module 2 (Credit 1)	
Learning Outcomes	1. Define and calculate measures of central tendency and dispersion.
	0. Interpret and compare datasets using summary statistics.
	0. Use spreadsheet functions to compute descriptive statistics from raw data.
	0. Calculate and interpret the correlation between two variables.
	0. Use spreadsheet functions to derive regression coefficients and trendlines.

Content Outline	**Descriptive Statistics** • Measures of central tendency (mean, median, mode) • Measures of dispersion (range, variance, standard deviation) • Percentiles, quartiles Using built-in spreadsheet functions: AVERAGE(), MEDIAN(), STDEV(), etc. **Correlation and Regression** • Correlation: Pearson coefficient • Simple linear regression • Using CORREL(), SLOPE(), INTERCEPT() Interpreting R^2 and trendlines
Module 3(Credit 1)	
Learning Outcomes	1. *Understand* various chart types and their appropriate use cases for different data sets. 2. Apply bar, pie, histogram, scatter, and line charts using spreadsheet tools.
	3. Apply Customize charts by adding titles, labels, legends, and colors for improved clarity.
	4. Understand key concepts in probability, including theoretical and empirical probability
	5. Apply simple probabilistic events using spreadsheet functions.
Content Outline	**Data Visualization** • Types of charts: bar, pie, histogram, scatter, line • Creating and customizing charts in spreadsheets • Interpreting graphical summaries • Chart design best practices **Probability and Distributions**

	• Basic probability concepts (theoretical & empirical) • Random variables, probability rules • Common distributions: Binomial, Normal, Poisson Spreadsheet functions: RAND(), BINOM.DIST(), NORM.DIST(), etc.
Module 4(Credit 1)	
Learning Outcomes	1. Understand the role of sampling and the importance of sample size in statistical analysis.
	2. Identify and apply appropriate sampling methods for different scenarios.
	3. Construct and interpret confidence intervals using spreadsheet tools.
	4. Formulate hypotheses and perform basic hypothesis tests (t-test, z-test, chi-square).
	5. Draw conclusions from test results and evaluate statistical significance.
Content Outline	**Inferential Statistics** • Sampling methods • Confidence intervals • Hypothesis testing: t-test, z-test, chi-square (basic) • Spreadsheet functions: T.TEST(), Z.TEST(), CHISQ.TEST()
Activities to be done in the class towards Comprehensive Continuous Evaluation	
Module 1: Activity 1: Classify the Data Activity 2: Real-World Data Collection Activity 3: Spreadsheet Scavenger Hunt Activity 4: Data Entry & Cleaning Task	

Activity 5: Create a Mini Dataset + Summary Table
Module 2: Activity 1: Calculate Central Tendency Activity 2: Analyze Dispersion Activity 3: Summary Table with Functions Activity 4: Visual Relationship with Scatter Plot Activity 5: Compute Pearson Correlation Activity 6: Build a Linear Regression Model
Module 3: Activity 1: Create Different Chart Types Activity 2: Visual Summary Dashboard Activity 3: Chart Comparison Analysis Activity 4: Simulate Random Events Activity 5: Explore Theoretical vs Empirical Probability Activity 6: Binomial Distribution Simulation
Module 4: Activity 1: Sampling Methods Exploration Activity 2: Construct a Confidence Interval Activity 3: One-Sample t-Test Activity 4: Two-Sample z-Test or t-Test Comparison Activity 5: Chi-Square Test of Independence

The Rubric will have the following Evaluation Parameters:		
Evaluation Parameters	**Description / Evaluation Points**	**Marks**
Introduction to Statistics & Spreadsheets	**Correct classification of data, explanation of collection methods, accurate use of basic functions**	**5**
Descriptive Statistics	**Accurate calculations using functions (AVERAGE, STDEV, etc.), correct interpretation**	**8**

Correlation and Regression	**Correct calculation of correlation & regression values, clear explanation of results/trendlines**	**8**
Data Visualization	**Chart selection, creation, customization, and interpretation; design best practices followed**	**8**
Probability and Distributions	**Simulations, use of functions, probability calculation, and matching distributions**	**8**
Inferential Statistics	**T.TEST, Z.TEST, CHISQ.TEST results with hypothesis interpretation and sampling justification**	**8**
Attendance and Conduct	**Regularity, discipline, and active participation**	**5**

Reference Books:

1. Statistics for People Who (Think They) Hate Statistics: Using Microsoft Excel" Author: Neil J. Salkind
2. Business Statistics: A First Course" (with Excel Guide)
3. Authors: David M. Levine, Timothy C. Krehbiel, Mark L. Berenson
4. Statistics for Managers Using Microsoft Excel"
5. Authors: David M. Levine, David Stephan, Kathryn A. Szabat

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

Course Title	**Introduction to Cyber Security**
Course Credits	**4**
Course Outcomes	After Completion of this Course, students will be able to
	1. Understand basic concepts of Cyber security
	2. Analyze different types of cyber attacks
	3.Implement Basic network Security Measures.
	4. Understand the concept of cryptography and secure communication over the Internet.
	5.Understand the concept of Malicious software
Module1 (Credit1): Fundamentals of Cyber Security	
Learning Outcomes	After learning this module, learners will be able to
	1. Understand the basic security principals.
	2. Understand the concepts of the CIA Triad.
	3. Understand all System and Network Security Concept.
	4. Aware about all the malwares.
Content Outline	Introduction to Cyber Security: Introduction to Cyber Security, History, Goals, Need of Security, Principles, Type of all Cyber Attacks, Security Policies(IT Act 2000,cyber Law),Security Techniques, Steps for Better Security, Basics Security Terminology (Cryptography, Hacking, Encryption,

	Decryption,Cryptanalysis,Cryptanalyst,Stegnography),Threats and Vulnerability,Difference between Security and Privacy. Malicious Software: Trojans,Backdoors, Viruses, and Worms, Trojans and Backdoors, Types of Trojans, Types of Viruses. Network & System Security: Public Key Infrastructure (PKI),Firewall,IDS(Intrusion Detection System),IPS(Intrusion Prevention System),Kerberos,Digital Signatures,Digital Certificates,Virtual Private Network(VPN). Internet Security Protocol: Basic concepts, SSL, SHTTP, TSP, SET, SSL versus SET, 3D secure protocol,Electronic Money, Email security, WAP security.
Module 2 (Credit1): Cryptography Basics	
Learning Outcomes	After learning this module, learners will be able to
	1.Understand Cryptographic Fundamentals.
	2.Apply Symmetric and Asymmetric Cryptography.
	3.Evaluate Cryptographic Algorithms.
	4.Explore Cryptographic Attacks.
Content Outline	Cryptographic techniques: Plaintext and Ciphertext,Substitution Techniques, Transposition Techniques,Encryption, decryption, Symmetric, Asymmetric Key Cryptography. Substitution Ciphers & Transposition Ciphers: Caesar, Monoalphabetic,Playfair, Hill, Polyalphabetic, One-time Pad or

	Vernam,Single Columnar,Rail Fence Cipher. Symmetric & Asymmetric Cryptographic Algorithms: Data Encryption Standard (DES),Advanced Encryption Standard (AES),Blowfish,RC5. RSA (Rivest-Shamir-Adleman),Diffie-Hellman Key Exchange,DSA (Digital Signature Algorithm). Authentication of People: Introduction, Authentication Methods(Password-based, Multi-Factor, Biometric,Verification techniques, passwords, length of passwords, smart cards).
Module 3 (Credit 1)	**Introduction to Public Key Cryptography and Authentication Mechanisms**
Learning Outcomes	After learning this module learners will be able to understand
	Understand the fundamentals of public key cryptosystems
	Describe and analyze the RSA algorithm
	Compare Kerberos with other security protocols
	Comprehend various authentication methods
Content Outline	Public Key Cryptosystems: Introduction, Public Key Cryptography, RSA Algorithm (Working of RSA, Key length and Security) Authentication: Introduction, Authentications Methods(Password-based, Two-factor, Biometric, Extensible). Kerberos: Basics, Ticket Granting Approach, Public Key Cryptography, Advantges, Weakness and attacks, Applications and Limitations, Comparison of

	Kerberos with SSL, Authentication Servers
Module 4 (Credit 1)	**Digital Signatures, Email Security, IP Security, and Web Security Protocols**
Learning Outcomes	Learners will be able to understand
	Analyze algorithms used for digital signatures
	Understand electronic mail security mechanisms
	Describe IP security concepts and architecture
	Understand web security protocols and their components
Content Outline	Digital Signatures: Introduction, Implementation, Association of Digital Signatures and Encryption, Using Different Key pairs for Signing and Encryption. Algorithms for Digital Signature: DSA(Digital Signature Algorithm, ECDSA(Elliptic Curve Digital Signature Algorithm), DSS, Attacks on Digital Signature. Electronic Mail Security: Introduction, Pretty Good Privacy (PGP), MIME, S/MIME, Comparison of PGP and S/MIME. IP Security: Introduction, IP Security Architecture, IPv6, IPsec, IPv4 and IPv6, IPsec Protocols and Operations Web Security: Introduction, SSL, SSL Session and Connection, SSL Record Protocol, Secure Electronic Transaction.
Activities to be done in the class towards Comprehensive Continuous Evaluation	
Module 1	

1.Define Cyber Security and explain its importance in today's digital world.

2.Briefly describe the history of Cyber Security and its evolution over time.

3.What are the goals of Cyber Security? Explain with examples.

4.List and explain the different types of Cyber Attacks with suitable examples.

5.Explain the following security terminologies with examples:

a) Cryptography

b) Hacking

c) Encryption

d) Decryption

e) Cryptanalysis

f) Cryptanalyst

g) Steganography

h) IT Act 2000 & Cyber Law

6.Define threats and vulnerabilities in Cyber Security.

7.List and explain different types of Trojans and their functionalities.

8.What are the various types of viruses? Explain any three with examples.

9.What is Public Key Infrastructure (PKI)? Explain its components and working.

10.Define Firewall and explain how it protects a network from cyber threats.

11.Write a short note on Intrusion Detection System (IDS) and Intrusion Prevention System (IPS).

12.What is Kerberos? How does it enhance network security?

13.Explain the role of Digital Signatures in cybersecurity.

14.How does a Virtual Private Network (VPN) work?

15.Define all Internet Security Protocols and explain their significance .

Module 2

1.Define Plaintext and Ciphertext with suitable examples.

2.Explain the difference between Substitution Techniques and Transposition Techniques in cryptography.

3.What is the difference between Encryption and Decryption? Provide real-world examples.

4.Compare Symmetric and Asymmetric Key Cryptography with examples.

5.Examples on Every Substitution and transposition techniques.

6.Explain all Symmetric algorithms in detail.

7.Explain all Asymmetric algorithms in detail.

8.Define authentication and explain its importance in cybersecurity. What are the different authentication methods used in security systems?

9.Compare password-based authentication with multi-factor authentication (MFA). Why is password length and complexity important for security?

10.Discuss the role of biometric authentication and smart cards in cybersecurity, along with real-world applications.

11.What are some password-cracking techniques, and how can users protect themselves? Explain a real-world case where MFA prevented a cyber attack.

Module 3:

1.Generate your own RSA keys using small prime numbers. Walk through the steps of key generation, encryption of a sample message, and decryption to retrieve the original message.

2.Find an example of how RSA is used in securing an online service or communication platform. Prepare a short presentation explaining its role.

3.Create a set of passwords and use online tools or scripts to test their strength. Discuss the features that make a password strong or weak.

4.Draw a detailed flow diagram of the Kerberos authentication process showing ticket granting and ticket usage.

Module 4:

1.How do digital signatures ensure both data integrity and non-repudiation?

2.Compare and contrast DSA and ECDSA. What are the advantages of elliptic curve algorithms?

3.Explain how IPsec enhances security in IPv4 and IPv6 networks.

4.Discuss how IPsec protocols (AH and ESP) differ in their security functions.

5.Describe the SSL handshake process. Why is it important for secure web communications?

6.What are the main security goals of the Secure Electronic Transaction protocol?

Reference Books:

1. Atul Kahate, Cryptography and Network Security, McGraw Hill.

2. William Stallings' "Cryptography and Network Security: Principles and Practice" is the 8th edition, published by Pearson Education Limited on June 6, 2022.
3. "Network Security: Private Communication in a Public World" is the 3rd edition, published on August 31, 2022.
4. "Cryptography and Information Security" by V.K. Pachghare, published in 2019.
5. "Cybersecurity Fundamentals: A Real-World Perspective" by Kutub Thakur and Al-Sakib Khan Pathan,published on June 30, 2020.
6. ”Computer Security: Principles and Practice (4th Edition)” by William Stallings and Lawrie Brown
7. Cybersecurity and Cyberwar: What Everyone Needs to Know by P.W. Singer and Allan Friedman

Assessment:

Internal Assessment: (Marks 50)

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	Clear definitions and explanations of cybersecurity concepts (e.g., RSA, Kerberos, Digital Signatures, IPsec). Demonstrates strong subject clarity and grasp of protocols and algorithms.	10
Accuracy and Completeness	All questions/activities are attempted. Answers are factually correct. Diagrams (e.g., flow of Kerberos tickets), algorithms, or cryptographic steps are accurately represented.	15
Application and Analysis	Correct application of concepts in practical scenarios (e.g., RSA key generation, attack analysis, email security protocols). Includes real-life relevance and security implications.	10
Presentation and Neatness	Well-organized layout with proper use of headings, indentation, bullet points, tables, and clearly labeled diagrams or screenshots of tools (e.g., encryption outputs).	5
Innovation and Effort	Attempts original examples or case studies. Uses practical demonstrations like protocol simulation, screenshots of	5

	software tools, or explains cybersecurity cases practically.	
Timely Submission	Submitted within the deadline. Late submissions incur up to 5 marks deduction unless exempted with valid reasons.	5

Project for Internal Evaluation

1.Identify and classify five real-world malware attacks (e.g., WannaCry, Stuxnet). Explain their impact and how they could have been prevented.

2.Create five strong passwords using best practices and explain why they are secure.

3.Write a step-by-step guide on configuring an SSL certificate for a website.

External Assessment: (Marks 50)

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

.1.4 Vocational Skill Courses (VSC)

Course Title	**Office Automation Tools**
Course Credits	**2 Credits**
Course Outcomes	**After learning the module, learners will be able to**
	1. Apply office automation concepts and technologies to integrate office tools and equipment, enhancing communication efficiency in the workspace.
	2. Analyse and evaluate proficiency in Writer and Calc, demonstrating mastery in text formatting, styles, graphics, tables, formulas, functions, and data analysis techniques.
	3. Design and manage complex documents and data using advanced features such as master documents, fields, forms, mail merge, data linking, collaboration tools, reviewing capabilities, and macros.
	4. Create and customize presentations in Impress, applying design principles to master slide creation, text and graphic formatting, slide transitions, animations, and various exporting options.
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Apply the concepts of office automation by integrating office tools, equipment, and technologies to optimize workspace communication and efficiency.
	2. Analyse and evaluate proficiency in using the Writer tool for text formatting, style application, graphics and table integration, mail merge operations, and document customization.
	3. Design and manage complex documents by applying advanced features like templates, master documents, fields, forms, and creating tables of contents, indexes, and bibliographies.

Content Outline	**Concept of Office Automation:** Purpose of an office, activities in an office ,structure of an office, office manual, document flow management in an office, need for office automation and its advantages and disadvantages, Office automation tools. **Office Automation Technology:** Office equipment, Workstation communication and convergence of Technologies **Writer** -Introducing Writer -Working with Text - **Formatting Pages** - Printing, Faxing, Exporting, and Emailing • **Introduction to Styles** - Working with Styles - **Working with Graphics** - Working with Tables – Working with Templates in Writer - Using Mail Merge – Creating Tables of Contents, Indexes, and Bibliographies - **Working with Master Documents** - Working with Fields - Using Forms in Writer- Customizing Writer
Module 2 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Apply Calc skills for proficient data management, analysis, and sharing, utilizing formulas, functions, and macros to enhance productivity.
	2. Analyse and evaluate proficiency in creating and customizing presentations in Impress, employing text and graphic formatting, slide transitions, animations, and various exporting options.
Content Outline	**Calc**: Introducing Calc, Entering, Editing, and Formatting Data, Using Charts and Graphs, Using Styles and Templates, Using Graphics in Calc, Printing, Exporting, and E-mailing, Formulas and Functions, Using the Data Pilot, Data Analysis, Linking Calc Data, Sharing and Reviewing, Calc Macros **Impress:** Guide Introducing Impress, Using Slide Masters, Styles, and Templates, Adding and Formatting Text, Adding and Formatting Pictures, Managing Graphic Objects, Formatting Graphic Objects, Spreadsheets, Charts, and Other Objects, Slides, Notes, and Handouts, Slide Shows : Transitions, Animations, Printing, Emailing, Exporting, and Saving SlideShows, Setting Up and Customizing Impress
Activities to be done in the class towards Comprehensive Continuous Evaluation	

Module 1:

• Research and identify three different office automation tools available in the market.

• Evaluate their advantages and disadvantages in terms of optimizing communication and efficiency in a workplace setting.

• Choose one tool and create a presentation using Impress to explain its features and benefits to your colleagues.

Module 2:

• Using Writer, create a complex document that includes text formatting, graphics, tables, and styles.

• Apply mail merge functionality to personalize documents for a hypothetical mailing list of 20 recipients.

• Evaluate the effectiveness of Writer's features for document customization and mail merge operations in a short reflective report (500 words).

• Design a comprehensive office manual template in Writer that includes sections for document flow management, office structure, and activities.

• Incorporate advanced features such as master documents, fields, and forms to create a dynamic and interactive document.

• Produce a table of contents, an index, and a bibliography within the manual template to demonstrate proficiency in document design and management.

References:

1. Weverka, P. (2015). Office 2016 All-In-One For Dummies. Wiley.
2. Libre Office Documentation Team. (2016). Libre Office: Writer, Calc, Math Formula Book. Friends of OpenDocument, Inc.
3. Leete, G., Finkelstein, E., & Leete, M. (2003). OpenOffice.org For Dummies. Wiley.
4. Weverka, P. (2018). Office 2019 All-in-One For Dummies. Wiley.
5. Libre Office Documentation Team. (2016). Libre Office 5.2 Writer Guide. Friends of OpenDocument, Inc.
6. Bain, M. A. (2006). Learn OpenOffice.org Spreadsheet Macro Programming: OOo Basic and Calc automation. A 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
Document Formatting (Word)	Consistently applies styles, headers, tables, bullets, TOC perfectly.	10
Spreadsheet Functions (Excel)	Uses formulas, charts, and functions (SUM, IF, VLOOKUP, etc.) accurately and effectively.	10
Presentation Skills (PowerPoint)	Slides are professional, well-designed, with animations and transitions effectively used.	10
Database Handling (Access or equivalent)	Well-structured tables, queries, forms, reports; relationships defined.	5
Presentation & Documentation	Project well-documented with clear screenshots, explanations, and references.	5

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

Project 1: "Small Business Management Toolkit"

Create a comprehensive Office Automation Toolkit to manage a small business (e.g., a bakery, bookstore, or tailoring service).

Project 2: "Event Planning System"

Create a digital system to manage all documentation and logistics for a college event (e.g., seminar, cultural fest, sports day).

.1.5 A. Skill Enhancement Courses (SEC)

Course Title	**Web Technology-I**
Course Credits	**2 Credits**
Course Outcomes	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 to be done in the class towards Comprehensive Continuous Evaluation

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:

External 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-on-similar sort of projects

Project 1: Using HTML/CSS design a single product page for an e-commerce site.

Project 2: Using HTML/CSS design a single-page resume builder using HTML forms.

1.5 B. Skill Enhancement Courses (SEC)

Course Title	**UI/UXI Design**
Course Credits	**2**
Course Outcomes	After Completion of this Course, students will be able to
	1. Understand the principles of User Interface (UI) and User Experience (UX) design
	2. Create wireframes, prototypes, and user personas based on research and usability.
	3.Apply design thinking methodologies to solve user-centric problems.
	4. Use industry-standard UI/UX tools for creating interactive designs.
Module1 (Credit1)	
Learning Outcomes	After learning this module, learners will be able to
	1. Explain the fundamentals of UI/UX and the difference between them.
	2. Conduct user research and define user personas.
	3. Design user flows, wireframes, and low-fidelity prototypes.
	4. Apply visual design principles in interface creation.
Content Outline	Introduction to UI/UX Design ● Difference between UI and UX ● Importance of UI/UX in product development ● Design Thinking and User-Centered Design ● UX Design Process Overview User Research & Analysis ● Methods of User Research (Interviews, Surveys, Observations)

	• Creating User Personas • Empathy Mapping • User Journey Mapping Wireframing & Prototyping • Sketching Techniques & Tools • Wireframe Design Principles • Low-Fidelity vs High-Fidelity Prototypes • Prototyping Tools (Figma, Adobe XD, Balsamiq) Visual Design Principles • Color Theory & Typography • Layout & Composition • Iconography & Imagery • Accessibility & Inclusive Design
Module2 (Credit1)	
Learning Outcomes	After learning the module, learners will be able to
	1. Develop interactive and responsive prototypes using UI tools.
	2. Apply usability testing methods to improve user experience.
	3. Understand the basics of responsive and mobile-first design.
	4. Evaluate and refine designs based on feedback and analytics

Content Outline	UI Design Tools & Interactive Prototyping • Introduction to Figma / Adobe XD • Components and Design Systems • Interactive Elements and Transitions • Designing for Multiple Devices Usability Testing & Feedback • Heuristic Evaluation • Usability Testing Methods • Gathering & Analyzing User Feedback • Iterative Design Process Responsive and Mobile-First Design • Responsive Design Principles • Grid Systems & Breakpoints • Designing for iOS and Android • Mobile Usability Best Practices Design Evaluation & Handoff • Preparing Design Specifications • Collaboration with Developers • Design Handoff using Zeplin/Figma Inspect • Design Documentation & Versioning

Reference Books

1. "Don't Make Me Think" – Steve Krug
2. "The Design of Everyday Things" – Don Norman
3. "About Face: The Essentials of Interaction Design" – Alan Cooper
4. "Lean UX: Designing Great Products with Agile Teams" – Jeff Gothelf
5. "UI is Communication" – Everett McKay

Practical Assignments

Sr. No	Practical Task Description	Module
1.	Conduct user interviews and surveys to create user personas and empathy maps	Module 1
2.	Create a user flow and wireframe for a mobile or web application	Module 1
3.	Design a low-fidelity prototype using Figma or Adobe XD	Module 1
4.	Apply color theory and typography to redesign an existing UI screen	Module 1
5.	Design a responsive UI mockup with interactions for a chosen app or website	Module 2
6.	Conduct usability testing with at least 3 users and summarize feedback	Module 2
7.	Refine your design based on user feedback and prepare handoff files	Module 2
8.	Present final prototype and document key UI/UX decisions	Module 2

Tools / Software Recommendations

Category	Tools/Software
UI/UX Design	Figma, Adobe XD, Balsamiq
User Research	Google Forms, Typeform, Miro, Whimsical
Testing & Feedback	Maze, Hotjar (Optional), Manual Testing
Design Handoff	Zeplin, Figma Inspect

The Rubric will have the following

1. User Research & Personal development	10
2. Wireframing & User Flows Structure, clarity, and completeness of wireframes and flow diagrams.	10
3. Visual Design & Prototyping Use of layout, color, typography, and interactive prototype in Figma/XD.	10

4. Usability Testing & Refinement Testing process, user feedback incorporation, and iteration.	10
5. Presentation & Documentation	10

Assessment:

External Assessment – 50 Marks

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

Semester - II

.2.1 Major (Core)

Course Title	**Data Structure & Algorithms**
Course Credits	**4 credits**
Course Outcomes	**1.** Apply abstract data structures using arrays and linked lists.
	2. Analyze the efficiency and performance of algorithms.
	3. Evaluate data structures like arrays, linked lists, stacks, queues, binary trees, and graphs for different applications.
	4. Design and apply appropriate searching and sorting techniques.
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	Apply various operations on linear data structures such as arrays, linked lists, stacks, and queues.
	Analyze algorithm characteristics, including space and time complexity, using asymptotic notation.
	Evaluate the advantages and disadvantages of different types of linked lists and their operations.
	Design and implement stack and queue structures, and apply their operations in different contexts.
Content Outline	**Algorithm Analysis:** Algorithm Characteristics, Space complexity, Time complexity. Asymptotic notation (Big O, 0, Omega and Theta) **Introduction:** Introduction to data structure, Classification of data structure, Operations performed on data structures, Linear data structure, arrays, operations on an array. **Linked List:** Introduction, Key terms, Advantages & disadvantages, Linear linked lists - Types (Singly, Doubly, Circular) Operations (Inserting, Deleting nodes) **Stack:** Introduction, Stack implementation, Operations on stack (Push Pop), Implementation of stack using pointer, Applications of stack, Infix prefix, postfix notations **Queue:** Introduction and Queue implementation, Operations.
Module 2 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	Apply various traversal techniques to binary trees and graphs.
	Analyze different sorting algorithms such as Bubble Sort, Quick Sort, and Heap Sort.

	Evaluate the efficiency of search algorithms like Linear Search and Binary Search.
	Design binary trees and graphs using array and linked list representations.
Content Outline	**Trees:** Introduction, terminology, Binary tree, Strictly Binary tree, Complete Binary tree, Binary tree representation as Array and Linked lists, Traversal (Inorder, Preorder, Post order), Binary Search Tree **Graphs:** Introduction, terminology, Graph representation, Applications of graph, Graph traversal (BFS, DFS, Shortest p **Sorting & Searching:** Bubble Sort, Selection Sort, Quick Sort, Heap Sort, Insertion Sort. Linear Search , Binary Search
Module 3 (Credit 1) **Data Structure Practical Approach**	
Learning Outcomes	**After learning the module, learners will be able to**
	Apply various operations on linear data structures such as arrays, linked lists, stacks, and queues.
	Analyze algorithm characteristics, including space and time complexity, using asymptotic notation.
	Evaluate the advantages and disadvantages of different types of linked lists and their operations.
	Design and implement stack and queue structures, and apply their operations in different contexts.
Content Outline	**Arrays:** Implementations of Array and Operations- Insertion, deletion of an element from one dimensional array, Traversing of array **Linked List:** Implementation of List and Linked List and Operations- Inserting, Deleting of nodes etc **Stack:** Stack Implementation, Operations on stack (Push Pop). Implementation of stack **Queue:** Implementation of Queue Implementation,Operations on queue(Insertion and deletion)
Module 4 (Credit 1)	

Learning Outcomes	**After learning the module, learners will be able to**
	Apply various traversal techniques to binary trees and graphs.
	Analyze different sorting algorithms such as Bubble Sort, Quick Sort, and Heap Sort.
	Evaluate the efficiency of search algorithms like Linear Search and Binary Search.
	Design binary trees and graphs using array and linked list representations.
Content Outline	**Trees** Implementation of tree as Array and Linked lists and Traversal (Inorder, Preorder, Postorder) **Graphs** Implementation of Graph traversal (BFS, DFS) **Sorting and Searching** Implementation of searching (Sequential, Binary search) Sorting (Bubble sort, Selection sort, Insertion Sort.)

Activities to be done in the class towards Comprehensive Continuous Evaluation

Module 1 : Implementing and Analyzing Linear Data Structures

Implement the following linear data structures using your preferred programming language:

Array
Singly Linked
List Doubly
Linked List
Circular Linked
List Stack
Queue

Write functions for basic operations on these data structures:

For arrays: insertion, deletion, and traversal.
For linked lists: insertion, deletion, and
traversal. For stacks: push and pop.
For queues: enqueue and dequeue.

Algorithm Analysis:

Analyze the time complexity of the implemented operations using asymptotic notation (Big O, Omega, Theta).

Module 2: Traversal and Sorting Techniques

1. **To apply traversal techniques to binary trees and graphs, analyze sorting algorithms, evaluate search algorithms, and design tree and graph structures.**
2. **Implement the following types of binary trees:**

   Binary Tree

   Strictly Binary Tree

   Complete Binary Tree

   Represent these trees using arrays and linked lists.
3. **Implement the traversal methods for binary trees:**

   Inorder Preorder Postorder
4. **Implement graph representations using adjacency matrix and adjacency list. Implement graph traversal algorithms:**

   Breadth-First Search (BFS) Depth-First Search (DFS)
5. **Sorting Algorithms:Implement the following sorting algorithms:**

   Bubble Sort Selection Sort Quick Sort Heap Sort Insertion Sort
6. **Search Algorithms:Implement Linear Search and Binary Search.**

Module 3:

Implementing and Analyzing Linear Data Structures

1. **Implement the following linear data structures using your preferred programming language:**

   Array

   Singly Linked

   List Doubly

   Linked List

   Circular Linked

   List Stack

   Queue
2. **Write functions for basic operations on these data structures:**

   For arrays: insertion, deletion, and traversal.

   For linked lists: insertion, deletion, and

   traversal. For stacks: push and pop.

   For queues: enqueue and dequeue.
3. **Algorithm Analysis:**

   Analyze the time complexity of the implemented operations using asymptotic notation (Big O, Omega, Theta).

Module 4

Traversal and Sorting Techniques

1. **To apply traversal techniques to binary trees and graphs, analyze sorting algorithms, evaluate search algorithms, and design tree and graph structures.**
2. **Implement the following types of binary trees:**

   Binary Tree

   Strictly Binary Tree

   Complete Binary Tree

   Represent these trees using arrays and linked lists.
3. **Implement the traversal methods for binary trees:**

   Inorder Preorder Postorder
4. **Implement graph representations using adjacency matrix and adjacency list. Implement graph traversal algorithms:**

   Breadth-First Search (BFS) Depth-First Search (DFS)
5. **Sorting Algorithms:Implement the following sorting algorithms:**

   Bubble Sort Selection Sort Quick Sort Heap Sort Insertion Sort
6. **Search Algorithms:Implement Linear Search and Binary Search.**

Reference Books

Trembley, P., & Sorrenson, R. (2005). Data Structure. McGraw Hill Publication.
Lipschuiz, S. (2014). Data structures (Schaum's Outline Series). McGraw Hill Publication.
Horowitz, E., & Sawhney, S. (2008). Fundamentals of Computer Algorithms. Computer Sci.P
Aho, A. V., Hopcroft, J. E., & Ullman, J. D. (1982). Data Structures and Algorithms. Pearson
Abhyankar, A. (2016) Data Structures and Files. Nirali Prakashan, Educational Publishers
Baluja, G. S. (2022). Data Structures Through C. VISIONIAS
Loomis, M. E. S. (1989). Data Management and File Structures (2nd ed.). Prentice Hall.
Samanta, D. (2009). Classical Data Structures. PHI, New Delhi.

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.	20
Practical Implementation	Competence in coding data structure operations in C, with focus on correctness, memory handling, and efficiency.	30
Problem Solving & Analysis	Ability to choose appropriate data structures, analyze complexity, and solve given problems effectively.	20
Design & Application	Skill in applying data structures to real-world problems (e.g., graphs, trees, queues), and implementing custom solutions.	10
Timely Submission	Submitted within the deadline. Late submission deducts up to 5 marks unless exempted.	20

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	**Computer Organization and Architecture**
Course Credits	**2 Credits**
Course Outcomes	Apply basic concepts to understand the structure of computers.
	Analyze the working of different interrupts and mapping techniques, and study register organization.
	Evaluate different addressing modes and their functionalities.
	Design and demonstrate the working of the central processing
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	Apply knowledge to understand the basic structure of a computer.
	Analyze memory organization and its components.
	Evaluate different memory types and their roles within a computer system.
	Design and implement memory organization techniques in computing scenarios.
Content Outline	**Basic Structure of computers:** Basic organization of computer, Intel 8086 Architecture, Basic Measures of Computer Performance, CPU: Registers, Computer Function: Instruction Cycle, Interrupts, Interconnection Structures, Bus Interconnection, Peripheral Component Interconnection (PCI). **Memory Organization:** Classifications of primary and secondary memories. Types of RAM (SRAM, DRAM, SDRAM, DDR, SSD) and ROM, Characteristics of memory, Memory hierarchy: cost and performance measurement. **Cache Memory:** Principles, Elements of cache design (Size, Mapping, Replacement, Write policies, Block size) Virtual Memory Concept.
Module 2 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	Apply various I/O techniques, including programmed I/O, interrupt-driven I/O, and direct memory access, to manage external devices and I/O modules.
	Analyze the instruction sets, operand types, operations on operands, and addressing modes of the 8086 processor.
	Evaluate the characteristics and performance differences between RISC and CISC architectures, including their pipelining capabilities.
	Design processor and register organizations based on the principles of
	RISC and CISC instruction execution.

Content Outline	**Input/Output:** External devices, I/O Modules, Programmed I/O, Interrupted-Driven I/O, Direct Memory Access. **Central Processing Unit:** Instruction sets: Instruction characteristics, Types of operands, Types of operations on operands, addressing modes of 8086 processor, Processor Organization, Register organization. **RISC:** Instruction Execution, RISC Characteristics, and RISC Pipelining, RISC Vs. CISC, Reduced Instruction Set Computers (RISCs), Introduction to CISC. CISC Characteristics
Activities to be done in the class towards Comprehensive Continuous Evaluation	
Module 1: 1. Study the basic organization of a computer system, focusing on Intel 8086 Architecture. 2. Explore the fundamental measures of computer performance and the role of CPU registers in the instruction cycle. 3. Classify primary and secondary memories, and analyze their classifications and characteristics. 4. Evaluate different types of RAM (SRAM, DRAM, SDRAM, DDR) and ROM, considering their cost and performance metrics. 5. Study the principles of cache memory. 6. Design elements of cache memory, including size, mapping, replacement policies, write policies, and block size considerations. 7. Discuss the concept of virtual memory and its implementation in modern computing. 8. Design a memory hierarchy system that optimizes both cost and performance for a specific computing scenario. 9. Implement the designed memory organization techniques in a simulation environment using appropriate tools or programming languages.	
Module 2 : 1. Implement various I/O techniques such as programmed I/O, interrupt-driven I/O, and direct memory access (DMA) to manage external devices and I/O modules. 2. Analyze the effectiveness and suitability of each technique in different computing scenarios. 3. Analyze the instruction sets, including characteristics, types of operands, and operations on operands of the Intel 8086 processor. 4. Evaluate the addressing modes and their implications for program execution efficiency. 5. Compare and contrast RISC and CISC architectures in terms of instruction execution efficiency and complexity. 6. Evaluate the advantages and disadvantages of RISC pipelining and its impact on processor performance. 7. Design a processor architecture based on either RISC or CISC principles, considering instruction set design and register organization. 8. Discuss how the designed architecture enhances performance and efficiency in specific computing tasks.	

Reference Books

1. Stallings, W. (2013). Computer organization and architecture: Designing for performance (10th ed.). Pearson.

2. Hayes, J. P. (1988). Computer architecture and organization. McGraw-Hill.
3. Hall, D. V. (2005). Microprocessor and interfacing (2nd ed.). Tata McGraw-Hill.
4. Brey, B. B. (2009). The Intel microprocessors 8086/8088… (4th ed.). PHI.
5. Tanenbaum, A. S. (2016). Structured computer organization (6th ed.). Pearson.
6. Mano, M. (2007). Computer system architecture (3rd ed.). Pearson.
7. Hwang, K., & Briggs, F. A. (1986). Computer architecture and parallel processing. McGraw Hill.
8. Chaudhuri, P. P. (2004). Computer organization and design. Prentice Hall India.
9. Usha, M., & Shrikant, T. S. (2014). Computer system architecture and organization. Wiley 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	**Object Oriented Programming Using C++**
Course Credits	**2 Credits**
Course Outcomes	Apply object-oriented programming concepts in C++.
	Analyze problems and develop C++ programs to solve them.
	Evaluate the use of file input/output in C++.
	Design solutions using object-oriented programming principles in C++.
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	Apply fundamental programming concepts including variables, data types, control structures, functions, arrays, and objects.
	Analyze and implement object-oriented programming concepts like objects, classes, and defining functions and variables..
	Evaluate the use of object-oriented programming in solving programming problems.
	Design solutions using a combination of fundamental programming and object-oriented concepts.
Content Outline	**Evolution of OOP:** Advantages and disadvantages of OOP over its predecessor paradigms. **Characteristics of Object-oriented Programming:** Abstraction, Encapsulation, Data hiding, Inheritance, Polymorphism, Code Extensibility and Reusability, User defined Data Types. **C++Program** Structure, Simple Input/ Output Program, Program Comments, Identifiers, Literals, String, Character, Integer, Floating Point, Constants, Keywords, Data Types, Operators in C++, Control Structures in C++. **Object and Classes:** Core object concepts, Encapsulation, Abstraction, Polymorphism, Classes, Messages Association, Interfaces, Implementation of class in C++, C++ Objects as physical object, C++ object as data types constructor Object as function arguments. **Functions and Variables:** Functions: Declaration and Definition, Variables: Definition Declaration, and Scope, Dynamic Creation and Derived Data, Arrays and Strings in C++.
Module 2 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**

	Apply concepts of constructors, inheritance (including its types), and polymorphism in C++.
	Analyze file input/output handling in C++ and class templates
	Evaluate the effectiveness of constructors, inheritance, and polymorphism in solving programming tasks.
	Design solutions involving file input/output operations and the utilization of class templates in C++.
Content Outline	**Inheritance:** Concept of Inheritance, Derived class and base class, Types of Inheritance, Functions and Friend Functions. • **Constructors:** Multiple Constructors and Initialization, Using estructors to Destroy Instances. **Polymorphism:** Syntax for Operator overloading, overloading of unary and binary operators. **File input and output:** Reading a File, Managing I/O Streams, opening a File – Different Methods, Checking for Failure with File Commands **Class templates:** Implementing a class template, implementing class template member functions, Using a class template, Function template.
Activities to be done in the class towards Comprehensive Continuous Evaluation	
Module 1: 1. Develop a console-based application in C++ that demonstrates the implementation of fundamental programming concepts and object-oriented principles. 2. Create a C++ program that includes classes representing real-world entities (e.g., a student, a car, a bank account). 3. Implement basic functionality within these classes, such as setting and retrieving object attributes, defining member functions for data manipulation, and demonstrating encapsulation and abstraction. 4. Utilize inheritance to create derived classes that inherit properties and behaviors from base classes, showcasing the concept of code reusability. 5. Incorporate polymorphism by defining virtual functions and overriding them in derived classes to demonstrate runtime polymorphism. 6. Implement file input/output operations to store and retrieve data related to objects, showcasing the handling of persistent data using C++. 7. Document your code thoroughly and provide comments to explain the purpose and functionality of each component.	

Module 2 : 1. Design and implement a template-based class hierarchy in C++ for managing a generic data structure. 2. Define a base template class that represents a generic data structure (e.g., a linked list, a stack, a queue). 3. Implement derived template classes that inherit from the base class and specialize it to 4. handle specific data types or functionalities (e.g., a linked list of integers, a stack of strings). 5. Utilize constructor overloading to provide flexibility in initializing instances of the template classes. 6. Implement operator overloading to enable intuitive manipulation of objects within the class hierarchy (e.g., addition, subtraction for mathematical operations).
1. Use file input/output operations to demonstrate the serialization and deserialization of objects of the template classes. 2. Test your implementation with various data types and scenarios to ensure correctness and functionality. 3. Provide comprehensive documentation explaining the design decisions, implementation details, and usage instructions for the template-based class hierarchy

Reference books:

1. Balaguruswamy, E. (2008). Object Oriented Programming with C++. Tata McGraw–Hill
2. Venugopal, K. R. (1997). Mastering C++. Tata McGraw-Hill Education.
3. Stroustrup, B. (1997). C++ Programming Language (3rd ed.). Addison Wesley.
4. Chandra, B. (1998). A Treatise On Object Oriented Programming using C++. Narosa Publications.
5. Schildt, H. (2001). The Complete Reference CN. Tata McGraw-Hill.

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**

Evolution & Concepts of OOP	Understanding of OOP evolution, advantages/disadvantages vs procedural programming. Clear explanation of abstraction, encapsulation, inheritance, polymorphism, and reusability.	10
C++ Program Structure	Demonstrates correct program structure with comments, proper use of identifiers, literals, keywords, data types, and operators.	10
Input/Output and Control Structures	Correct use of input/output (cin, cout), conditional statements (if, switch), and loops (for, while, do-while).	10
Classes and Objects	Correct implementation of classes, constructors, destructors, objects, member functions, and passing objects as arguments. Applies encapsulation and abstraction.	10
Inheritance	Accurate implementation of inheritance (single, multiple, multilevel, hierarchical), correct base and derived class usage, and understanding of access specifiers.	10

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

Project 1: Employee Payroll Management System

A payroll system that handles different types of employees, calculates salaries based on employee type (Full-time, Part-time, Contract), and generates reports using file handling.

Project 2: Student Report Card Management System

Create a system that manages student records, calculates grades, and saves report data using OOP principles and file handling.

.2.4 Vocational Skill Courses (VSC S2)

Course Title	**Linux Operating System**
Course Credits	**2**
Course Outcomes	**After going through the course, learners will be able to**
	1. Apply Linux environment skills effectively for various tasks and operations.
	2. Analyse and utilize basic Linux commands and shell scripting techniques.
	3. Evaluate file system creation, directory management, and related operations using Linux programs.
	4. Design and implement solutions using a diverse set of standard Linux programming and development tools.
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Apply basic Linux commands proficiently for various tasks and operations.
	2. Analyse the significance of Linux architecture and its key features.
	3. Evaluate Linux utilities for creating and managing simple file processing operations effectively.
	4. Design and demonstrate memory management techniques in file handling, including file and region locking mechanisms.
Content Outline	**Introduction to Linux** Introduction to Unix architecture, General-purpose utilities , All basic commands introduction,Usage of commands. **Files and Directories:** File Concept, File types, File System Structure, File metadata- Inodes, kernel support for files, file System calls for file I/O operations- open, create, read, write, close. **Directories-** mkdir, rmdir, chdir, obtaining current working directorygetcwd, directory contents, scanning directories opendir, readdir, closedir, rewind dir functions.
Module 2 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Apply the concept of filters in Linux to manipulate data

	effectively.
	2. Analyse different editors available in Linux and their functionalities.
	3. Design and implement shell scripts capable of performing complex tasks in a shell programming environment.
	4. Evaluate the role and functionalities of the Linux kernel in system operations.
Content Outline	**The vi Editor** Introduction to the vi editor of Linux.all commands. **Introduction to filters**
	Simple Filters, Filters using regular expressions - use of grep command **Introduction to shell concept and writing shell script-** What is kernel, What is Shell,what is terminal,Advantage and disadvantage, first simple Shell Programme.

Activities to be done in the class towards Comprehensive Continuous Evaluation

Module 1:

1. Basic Linux Commands Proficiency:

Practice using a range of basic Linux commands for file and directory management, navigation, and system operations.

Apply these commands to perform tasks such as file creation, modification, deletion, and directory navigation.

2. Analysis of Linux Architecture:

Analyze the architecture of Unix/Linux systems, including its components and functionalities. Evaluate the significance of key features like multi-user capability, security models, and file system structure in Unix/Linux.

3. Evaluation of Linux Utilities:

Evaluate Linux utilities for file processing operations, including creation, deletion, copying, and searching files and directories.

Analyze the efficiency and effectiveness of these utilities in real-world scenarios.

4. Design and Demonstrate Memory Management Techniques:

Design a system for managing file memory, including techniques like file and region locking mechanisms.

Implement and demonstrate these techniques using Linux system calls and utilities.

Module 2:

1. Application of Filters:

Apply filters in Linux to manipulate data effectively, including simple filters and those using regular expressions.

Use commands like grep for text pattern searching and filtering.

2. Analysis of Linux Editors:

Analyze different editors available in Linux, focusing on the vi editor.

Explore and practice all essential commands and functionalities of the vi editor for text editing and manipulation.

3. Design and Implementation of Shell Scripts:

Design shell scripts capable of performing complex tasks in a shell programming environment. Develop scripts that automate file management, data processing, or system administration tasks.

4. Evaluation of Linux Kernel Functionalities:

Evaluate the role and functionalities of the Linux kernel in system operations, including process management, memory management, and device management.

Discuss the advantages and disadvantages of kernel-level operations and their impact on system performance.

Reference books:

1. Card, R., Dumas, E., & Mevel, F. (2003). The Linux kernel book. Wiley.
2. Das, S. (2017). Unix concepts and applications (4th ed.). TMH.
3. Suehring, S. (2002). MySQL Bible. John Wiley & Sons.
4. Lerdorf, R., & Tatroe, K. (2002). Programming PHP. O'Reilly Publications. 5. Collings, T., & Wall, K. (Wiley). (2003). Red Hat Linux network and system administration (3rd ed.).
5. Mathews, N. (2007). Beginning Linux programming (4th ed.). Wrox Press.
6. Koparkar, P. (2001). Unix for you. Tata McGraw-Hill.
7. Kanetkar, Y. P. (2013). Unix shell programming (1st ed.). BPB Publications.

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	All required Linux features and functions are implemented (file ops, filters, memory mgmt, shell scripts, vi editor, etc.) with full functionality	10
Code Structure & Best Practices	Code is modular, well-commented, follows naming conventions, and uses appropriate Linux commands and logic	10
User Interface & Design Consistency	Command-line interaction is user-friendly, intuitive, with helpful prompts and consistent format	10
Input Validation & Exception Handling	Validates all user inputs (e.g., file names, paths, user options); handles all exceptions gracefully using built-in checks or control structures	10
Technical Documentation	Professional report with clear structure: introduction, architecture, code snippets, output screenshots, and detailed explanation	10

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

Project 1: Linux Automation & Memory Safety Toolkit

A multi-functional script that automates system tasks (file handling, process monitoring, logs) and implements memory/file locking mechanisms.

Project 2: Linux Command Trainer with vi & Filters

Build an interactive command-line learning
environment for beginners. Create interactive
shell script tutorials that teach use of:
grep, awk, sed, cut, sort, etc. Regular expressions and filters
Build an in-terminal vi editor tutorial, explaining modes, commands, macros
Write a formal report, use screenshots, and prepare a demo presentation with voice-over

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

Course Title	**Multimedia System**
Course Credits	**4**
Course Outcomes	**After going through the course, learners will be able to**
	1. Apply creativity, organization, and communication effectively in project stages.
	2. Analyse and optimize text, images, sound, and video for different mediums.
	3. Evaluate mastery of multimedia software tools for image editing, sound editing, animation, and video production.
	4. Design efficient file management strategies using compression techniques like CODECs, GIF, JPEG, MPEG to control file sizes.
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Apply understanding of multimedia significance and project development stages.
	2. Analyse font types, utilize text editing tools, and design web specific multimedia content.
	3. Evaluate production of still images using bitmap and vector drawing techniques, and understanding of color theory and image file formats.
	4. Design application of digital audio concepts, video technology workings, and animation principles in multimedia projects effectively.

Content Outline	**INTRODUCING MULTIMEDIA:** Multimedia- Definitions, Use of Multimedia, Introduction To Making Multimedia: The Stages of a Multimedia Project, Need, Creativity, Organization, Communication. **Tex**t- About Fonts and Faces, Cases, Serif Versus Sans Serif, Using Text in Multimedia, Computers and Text, Font editing and design tools, Hypermedia and Hypertext. Designing for the World Wide Web-Developing for the Web, Text for the Web, Images for the Web, Sound for the Web, Animation for the Web. **IMAGES:** Images: Making Still Images, Bitmaps, Vector Drawing, 3-D Drawing and Rendering, Color, Understanding Natural Light and Color, Computerized Color, Color Palettes, Image File Formats.
Module 2 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Apply multimedia system sound principles, including digital audio, MIDI, and various audio file formats.
	2. Analyse differences between MIDI and digital audio technologies in multimedia contexts.
	3. Evaluate methods for audio CD playback, recording, and voice recognition systems.
	4. Design and implement video technologies, including broadcast standards, digital video formats, and techniques for optimizing video files and animations for multimedia applications.
Content Outline	**Sound:** Multimedia System Sounds, Digital Audio, MIDI Audio, Audio File Formats, MIDI vs Digital Audio, Audio CD Playback.Audio Recording. Voice Recognition & Response. **Video:** How Video Works, Broadcast Video Standards: NTSC, PAL, SECAM, ATSC DTV, Analog Video, Digital Video, Digital Video Standards – ATSC, DVB, ISDB, Video recording & Shooting Videos, Video Editing, Optimizing Video files for CD ROM, Digital display standards. **Animation**: Principle of Animations. Animation Techniques, Animation File Formats
Module 3 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Apply image editing techniques using selection tools, layers, masks, and channels in software applications.

	2. Analyse the differences between Macintosh and Windows hardware platforms, including connectivity options like IDE, SCSI, ATA, USB, and Firewire.
	3. Evaluate storage devices and multimedia input/output devices suitable for multimedia projects.
	4. Design strategies for producing web-ready files and integrating typographic design and vector drawing into multimedia content.
Content Outline	**IMAGE EDITING:** Image Editing software: selection tools, working with layers, masks and channels, correcting and enhancing photographs, typographic design and vector drawing, working with 3D images, producing files for the web. **Hardware:** Macintosh versus Windows, Connections: IDE, SCSI, UIDE, ATA, USB, Firewire etc. Storage devices, Input , Output devices for Multimedia Projects
Module 4 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Apply mastery of image editing software for diverse content creation.
	2. Analyse hardware differences to select appropriate devices.
	3. Evaluate proficiency in various multimedia software tools.
	4. Design effective application of compression principles for web development.
Content Outline	**Multimedia Software Tools:** Text Editing & Word processing tools, OCR S/W, Painting &Drawing Tools, 3D Modelling & Animation Tools, Image editing tools, Sound Editing tools, Animation, Video & Digital movie tools, Overview of various types of Multimedia Authoring tools. **Compression**: CODEC, Types of Compression & redundancies, GIF, JPEG & MPEG Standards Overview, Fractals **Multimedia tools for WWW & Designing for WWW:** Plug Ins, Text, Images, Sound & Animation for the Web
Activities to be done in the class towards Comprehensive Continuous Evaluation	
Module 1: **Multimedia Project Proposal:** Develop a proposal for a multimedia project, including objectives, target audience, and content outline. Present the proposal to the class, highlighting the need for creativity and effective communication.	

Font Analysis and Design:
Analyze different font types and their usage in multimedia. SNDTWU 2024 BSc(IT) Programme Structure as per NEP-2020
Design a multimedia presentation focusing on font selection and its impact on communication.

Module 2:
Image Creation and Editing Task:
• Create still images using bitmap and vector drawing techniques.
• Edit and enhance the images using image editing software, considering color theory and file formats.
Audio Recording and Editing Exercise:
• Record and edit audio clips using sound editing software.
• Explore various audio file formats and techniques for enhancing audio quality.

Module3:
Video Production and Optimization Project:
• Produce a short video clip, applying principles learned about video production, editing, and optimization.
• Optimize the video for CD-ROM distribution and digital display standards.
Image Editing and Web Production Task:
• Use image editing software to produce web-ready graphics, focusing on selection, layering, and optimization techniques.
Hardware and Device Comparison:
• Research and compare Macintosh and Windows systems, as well as various hardware connections and storage devices.
• Present findings in a comparative analysis report.

Module 4:
Multimedia Software Exploration:
• Explore various multimedia software tools, including text editing, painting, 3D modeling, animation, and sound editing.
• Create a multimedia project using a combination of these tools.
Compression Techniques Analysis:
• Investigate different compression techniques and standards such as CODECs, GIF, JPEG, MPEG, and Fractals.
• Compare and contrast techniques and present findings in a multimedia presentation.

Reference Books:

1. Buford, J. F. K. (2002). Introduction to Multimedia Systems.Pearson.
2. Vaughan, T. (1999). Introduction to Multimedia. McGraw-Hill Osborne Media
3. Gonzalez, R. C., & Woods, R. E. (2018). Digital Image Processing.Pearson.
4. Pratt, W. K. (1991). Digital Image Processing. John Wiley & Sons
5. Adobe Creative Team. (2021). Adobe Photoshop Classroom in a Book. 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**
Core Functionality & Feature Coverage	Implements all multimedia features (e.g., image/audio/video processing, compression, codecs, GUI) correctly and completely	10
Code Structure & Best Practices	Modular, reusable code with good documentation, follows design patterns and clean coding practices	10
User Interface & Design Consistency	Attractive and intuitive UI/UX; consistent layout, design aesthetics, and responsive behavior	10
Input Validation & Exception Handling	Validates multimedia input types (image/audio/video), prevents crashes, and handles exceptions gracefully	10
Technical Documentation	Well-organized report: introduction, methods, screenshots, technical details, user manual	10

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

Project 1: Mini Multimedia Player with Custom Codec Support

Create a basic multimedia player that supports multiple audio/video formats and demonstrates codec implementation.

Project 2: Image & Video Compression Toolkit

Build a tool that demonstrates compression techniques and shows quality-size trade-offs visually.

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 Artificial Intelligence**
Course Credits	**4 Credits**
Course Outcomes	1. Understand the basic concepts and applications of Artificial Intelligence.
	0. Describe various types of intelligent agents and analyze the characteristics of their operational environments.
	0. Formulate and solve Constraint Satisfaction Problems, including real-world examples like map colouring and scheduling.
	0. Integrate knowledge from previous modules to design and demonstrate simple AI applications.
	0. Critically evaluate ethical dimensions of AI systems, including bias, fairness, transparency, privacy, and explainability.
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Explain the significance and application of data in real-world contexts
	2. Explain fundamental AI concepts, including definitions, objectives, and major historical milestones
	3. Classify the different types of agents
	4. Analyze and classify task environments (fully vs. partially observable, deterministic vs. stochastic, episodic vs. sequential, static vs. dynamic, single-agent vs. multi-agent) and assess how these affect agent design.

Content Outline	Definitions and applications of AI: problem-solving, learning, perception, reasoning Historical milestones: Turing Test, Dartmouth workshop, expert systems, emergence of ML and deep learning Agent architectures: reflex, model-based, goal-oriented, utility-based agents Task environments: fully/partially observable, deterministic vs. stochastic, episodic vs. sequential, static vs. dynamic, single vs. multi-agent.
Module 2 (Credit 1)	
Learning Outcomes	*1.* Explain and differentiate between uninformed and informed search strategies, including BFS, DFS, Uniform Cost Search, Greedy Best-First Search, and A, with respect to their applications and performance.
	2. Apply uninformed and informed search algorithms to solve problems such as pathfinding and puzzle-solving, demonstrating an understanding of their strengths and limitations.
	3. Formulate real-world problems as Constraint Satisfaction Problems (CSPs), identify variables, domains, and constraints, and apply appropriate techniques to find valid solutions.
	4. Analyze adversarial game scenarios using the Minimax algorithm and enhance efficiency using alpha-beta pruning, understanding how these methods optimize decision-making in competitive environments.
	5. Design and interpret flowcharts representing different search strategies, highlighting the algorithmic flow and decision points within each approach.
Content Outline	**Search, Planning & Problem-Solving** • Uninformed search: BFS, DFS, Uniform Cost Search • Informed search: Greedy Best-First, A* algorithm • Constraint satisfaction problems (CSPs): definitions, examples (e.g., scheduling, map coloring) • Adversarial games: Minimax, alpha-beta pruning **Activities:** Puzzle-solving lab, planning exercises, flowcharts of search strategies

Module 3(Credit 1)	
Learning Outcomes	1. *Understand* the basic concepts of machine learning.
	2. Understand and distinguish between supervised, unsupervised, and reinforcement learning, identifying when each paradigm is appropriate.
	3. Compare machine learning algorithms in terms of their assumptions, strengths, limitations, and performance across different problem types
	4. Explain the different application and ethical aspects.
Content Outline	**Machine Learning Foundations** ● Overview: supervised vs unsupervised learning, classification vs regression ● Decision trees, naïve Bayes, k-means clustering, PCA basics ● Neural networks: architecture, perceptron, backpropagation ● Real-world applications and ethical aspects (fairness, bias, transparency)

Module 4(Credit 1)	
Learning Outcomes	1. Identify and describe key AI application areas—including Natural Language Processing (NLP), Computer Vision (CV), Robotics, and Expert Systems—and explain their real-world significance and use cases.
	2. Assess the ethical dimensions of AI, including fairness, bias, transparency, accountability, and privacy, particularly in sensitive applications like education and hiring.
	3. Apply ethical reasoning to case studies in AI, such as grading systems and recruitment tools, to identify risks and propose mitigation strategies .
	4. Understand the responsible AI practices, incorporating human-centered and transparency-driven methods into AI design,

	deployment, and governance
Content Outline	**Applications and Ethical AI** ● Overview of AI applications: NLP, CV, robotics, expert systems ● Basic overview of fuzzy logic, genetic algorithms ● Discussion on AI ethics, fairness, and societal impact
Activities to be done in the class towards Comprehensive Continuous Evaluation	
Module 1: Agent Simulation Workshop: Students simulate a vacuum-cleaner agent, defining its sensors, actuators, and decision-making logic. PEAS Model Exercise: Analyze real-world agents (like chatbots or smart home devices) using the PEAS (Performance, Environment, Actuators, Sensors) framework. Historical Timeline Group Activity: Create a visual timeline of AI milestones (Turing Test, Dartmouth conference, expert systems, deep learning breakthroughs).	
Module 2: Puzzle-Solving Lab: Implement and compare BFS, DFS, UCS, Greedy Best-First, and A* on puzzles like the 8-puzzle or maze navigation. CSP Workshop: Model scheduling or map-coloring problems as CSPs, applying techniques like backtracking and forward checking.	
Module 3: Feature Engineering Challenge: Students clean data, handle missing values, encode variables, and engineer features for modeling. Evaluate the impact with cross-validation and metrics like accuracy, precision, and recall. Ethics Debate: Teams debate real-world ML scenarios (e.g., facial recognition, loan approval) addressing privacy, fairness, and bias issues.	
Module 4: NLP: Build a simple chatbot using pre-trained language models. CV: Use OpenCV or ML libraries to detect objects or faces in images. Fuzzy Logic / Genetic Algorithm Demo: Create a basic fuzzy controller (e.g.,	

thermostat) or optimize a simple function using genetic algorithms.

The Rubric will have the following Evaluation Parameters:		
Evaluation Parameters	**Description / Evaluation Points**	**Marks**
Conceptual Understanding	Demonstrates deep, accurate comprehension of AI concepts, agents, search, ML, and ethics across all modules with clear connection	10
Technical Implementation & Accuracy	Produces flawless code and algorithms (search, CSP, ML, demo) with well-documented, efficient, logically structured implementation.	10
Critical Reasoning & Problem-Solving	Chooses and justifies appropriate AI techniques; compares alternatives; analyzes algorithmic trade-offs and optimization opportunities.	10
Ethical Awareness & Reflection	Critically evaluates fairness, bias, transparency, and privacy in AI; proposes concrete mitigation strategies supported by frameworks.	10
Communication & Presentation	Presents ideas clearly and professionally; visuals are polished; responds effectively to questions.	10

Text Book:

1. Artificial Intelligence (Third Edition) McGraw-Hill Elaine Rich, Kevin Knight.

Reference Books:

1. A First course in Artificial Intelligence (McGraw-Hill) Deepak Khemani.
2. Artificial Intelligence A modern approach (Second Edition) Pearson, Stuart Russell, and Peter
3. Norvig.
4. Fuzzy Logic with Engineering application (Third edition) Timothy J. Rose.
5. Artificial Intelligence and Intelligence system: N. P. Padhy
6. Artificial Intelligence: Patrick Henry Winston
7. Artificial Intelligence (Structure & Strategies for Complex Problem solving): George F. Luger

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

Course Title	**Cyber Forensic**
Course Credits	**2**
Course Outcomes	**After Completion of this Course, students will be able to**
	1. Understand the Fundamentals of Cyber Forensics
	2. Analyze Digital Evidence and Investigative Techniques
	3.Investigate Cybercrimes and Incident Response
	4. Apply Forensic Tools and Report Findings
	5. Understand the concept of legal considerations.
Module1 (Credit1):	
Learning Outcomes	**After learning this module, learners will be able to**
	1.Understand the fundamental concepts of Digital forensics.
	2. Understand the fundamental concepts of cyber forensics.
	3. Identify different types of digital evidence and legal considerations.
	4. Learn forensic methodologies and the role of forensic experts in investigations.
Content Outline	**Introduction to Digital Forensic:** Introduction to cyber crimes & Digital Forensic, Types of Digital Forensics, Digital Forensics Process, Areas of Application of computer forensics, Understanding the Suspects, Examples of

	Computer Forensics, Free space and Slack Space. **Introduction to Cyber Forensics** Definition, Scope, Importance of Cyber Forensics, Digital Crime Investigation Process, Cyber Laws and Ethics in Forensics **Types of Digital Evidence & Crime Scenes** Sources of Digital Evidence (Computers, Mobile Devices,Networks) ,Volatile vs. Non-Volatile Evidence ,Evidence Collection and Chain of Custody **Legal and Ethical Considerations in Cyber Forensics** Admissibility of Digital Evidence in Court,Role of Law Enforcement Agencies,Ethical Issues in Cyber Forensics
Module 2 (Credit1):Cybercrime Investigations & Forensic Tools	
Learning Outcomes	**After learning this module, learners will be able to**
	1.Gain knowledge of forensic tools used in digital investigations.
	2.Analyze forensic methodologies for network and mobile forensics.
	3.Apply forensic techniques to real-world cybercrime cases.
	4.Understand the concept of mobile & network forensic.
Content Outline	**Forensic Tools and Techniques:** Popular Forensic Tools: Autopsy, EnCase, FTK, Wireshark File System Forensics (FAT, NTFS) Recovering Deleted Files & Metadata Analysis **Email,Social Media,Network & Mobile Forensics:**

	Network Traffic Capture & Analysis (Wireshark, TCPDump) Email & Social Media Forensics,Exploring the role of email investigation,Applying Digital Forensics Methods to Social Media Communications, Social Media Forensics on Mobile Devices Forensics Tools for Social Media InvestigationsMobile Device Forensics (Android, iOS),Why do we need mobile forensics?, Challenges in mobile forensics. **Cybercrime Investigation & Case Studies** Investigating Hacking, Identity Theft, Phishing Attacks Financial & Banking Frauds (Ransomware, Cryptocurrency Fraud) Case Studies on Real-World Cybercrimes **Emerging Trends in Cyber Forensics** Cloud & IoT Forensics,AI & Machine Learning in Digital Forensics, Challenges in Dark Web & Deep Web Investigations
Activities to be done in the class towards Comprehensive Continuous Evaluation	
Module 1: 1.Define Cyber Forensics and explain its role in modern investigations. 2.Differentiate between volatile and non-volatile digital evidence. 3.Describe the Cyber Forensic Investigation Process. Explain each step in detail with a relevant case study. 4.What are the key challenges in digital evidence collection? 5.Explain the significance of the chain of custody in cyber forensics. 6.Discuss the role of cyber laws and ethics in digital forensics. 7.What are the different sources of digital evidence? Explain how evidence is collected from different devices like computers, mobile phones, and networks. 8.Analyze the importance of hashing in digital forensics. 9What is forensic imaging? Explain the different types of imaging techniques used in cyber forensic investigations. 10.Discuss the role of law enforcement agencies in cyber forensic investigations.	
Module 2	

1.Compare and contrast Autopsy, EnCase, FTK, and Wireshark.

2.Explain the role of file system forensics in digital investigations. How are FAT and NTFS file systems analyzed for forensic purposes?

3.How do forensic experts recover deleted files? Discuss the challenges and techniques used in metadata analysis.

4.Describe the process of capturing and analyzing network traffic using Wireshark and TCPDump.

5.How does email forensics help track cybercriminal activities? Explain the importance of email header analysis in forensic investigations.

6.What are the challenges in social media forensics?

7.Why is mobile forensics important in cyber investigations?

8. Explain key forensic techniques used in Android and iOS devices and the challenges faced in mobile forensic analysis.

9.Investigate a real-world cybercrime case involving hacking, identity theft, or phishing attacks.

10.Explain how forensic techniques were used to analyze the incident.

11.How do forensic experts investigate financial and banking frauds?

12.Discuss the challenges in cloud and IoT forensics.

13.How is artificial intelligence (AI) and machine learning (ML) transforming digital forensics?

14.Explain the term Dark Web and Deep Web?

15.Discuss digital forensic methods used for investigating social media communications and mobile devices.

References:-

1. "Guide to Computer Forensics and Investigations" by Bill Nelson, Amelia Phillips, and Christopher Steuart is the 7th Edition, published by Cengage Learning.
2. "The Basics of Digital Forensics: The Primer for Getting Started in Digital Forensics" by John Sammons is the Second Edition, published by Syngress,
3. Nilakshi Jain, Dhananjay Kalbande, ”Digital Forensic : The fascinating world of Digital Evidences ” Wiley India Pvt Ltd 2017.
4. Clint P Garrison “Digital Forensics for Network, Internet, and Cloud Computing A forensic evidence guide for moving targets and data , Syngress Publishing, Inc. 2010.
5. Cory Altheide, Harlan Carvey ”Digital forensics with open source tools “Syngress Publishing,Inc. 2011.

Assessment:

External Assessment: (50 Marks)

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

.2.6 Skill Enhancement Courses (SEC)

Course Title	**Mathematics -I**
Course Credits	**2 Credits**
	1. Identify and explain the fundamental concepts of sets.
	2. Understand and illustrate concepts of permutations, combinations, and graph theory.
	3. Apply logical reasoning to construct simple mathematical proofs and solve related problems.
	4. Demonstrate various graph traversal techniques and explain their use in different scenarios.
	5. Analyze and apply discrete mathematical structures such as relations, functions, and graph theory to solve computational problems.
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Students will demonstrate the ability to perform and apply operations on sets, such as union, intersection, complement, set difference, and Cartesian product, and will understand how these operations are visualized using Venn diagrams. 0. Students will be able to differentiate between permutations and combinations and apply the appropriate formulae to solve simple problems involving arrangements and selections of objects, demonstrating the practical applications of these concepts.
Course Outline	**Set Theory:** Definition of Sets, Subsets, Cardinality of Sets, types of sets: Equal Sets, Universal Sets, Finite and Infinite Sets, proper set, power sets, Operations on Sets: Union, Intersection, Complement of Sets, set difference, Cartesian Product, Venn Diagrams, and Algebra of sets **Permutations and Combinations:** Definitions: Permutation, Combination and simple problems

Module 2 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Students will be able to define matrices, perform various operations (addition, multiplication, and finding the inverse of square matrices), and compute determinants of matrices, applying properties of determinants to solve problems. 2. Students will be able to define and classify different types of graphs, apply concepts like the Handshaking Lemma, understand graph isomorphism, and work with subgraphs and the complement of graphs to solve related problems.
Course Outline	• **Matrices and Determinants** Definition of a matrix; Operations on matrices; Square Matrix and its inverse; determinants; properties of determinants; • **Graph theory** Graphs, types of graphs, Handshaking Lemma, Isomorphism of graphs, Subgraphs, Complement of graph.

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE) for Numerical Methods

Module 1:

1. Solve examples of set theory
2. An assignment in set theory could involve exploring the use of **Venn diagrams** to visualize complex set relationships.

Module 2:

1. Solve examples based on Matrices and Graph Theory
2. An assignment to model a city's transport system (like bus stops or metro stations) as a graph, represent it using matrices, and use determinants and matrix operations to analyze connectivity and efficiency.

Reference Books:

1. Kolman, Busby and Ross, "Discrete mathematical Structures and graph theory"
2. Alan Doerr, K. Levasseur, "Applied discrete structure for computer science", Galgotia publications, 1988

3. Trembley&Manohar, "Discrete mathematical Structures with application to computer science", McGraw Hill, 1987.

4. Swapan Kumar Chakraborty, BikashKantiSarkar, Discrete Mathematics, Oxford Higher Education, 2011

5. C. L. Liu, D. P. Mohapatra, Elements of Discrete Mathematics A Computer Oriented Approach, Tata Mcgraw-Hill, 3 rd Edition, 2008.S.

Assessment:

Internal Assessment – 50 Marks Evaluation Scheme:

Using C++ programming 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.	20
Practical Implementation	Competence in coding data structure operations in C, with focus on correctness, memory handling, and efficiency.	30
Problem Solving & Analysis	Ability to choose appropriate data structures, analyze complexity, and solve given problems effectively.	20
Design & Application	Skill in applying data structures to real-world problems (e.g., graphs, trees, queues), and implementing custom solutions.	10
Timely Submission	Submitted within the deadline. Late submission deducts up to 5 marks unless exempted.	20