SNDT Women's University, Mumbai

Bachelor of Science

(Computer Science)

B.Sc. (CS)

Syllabus

as per NEP 2020

w.e.f.

A.Y.: 2024-2025.

1, Nathibai Thackersey Road, Mumbai-400020

www.sndt.ac.in

Terminologies

Abbreviation	Full-form	Remarks	Related to Major and Minor Courses
Major (Core)	Main Discipline		
Major (Elective)	Elective Options		related to the Major Discipline
Minor Stream	Other Disciplines (Inter/ Multidisciplinary) not related to the Major	either from the same Faculty or any other faculty	
OEC	Open Elective Courses/ Generic		Not Related to the Major and Minor
VSEC	Vocational and Skill Enhancement Courses		
VSC	Vocational Skill Courses		Not Related to the Major and Minor
SEC	Skill Enhancement Courses		Not Related to the Major and Minor
AEC	Ability Enhancement Courses	Communication skills, critical reading, academic writing, etc.	Not Related to the Major and Minor
VEC	Value Education Courses	Understanding India, Environmental science/education, Digital and technological solutions, Health & Wellness, Yoga education, sports, and fitness	Not Related to the Major and Minor
IKS	Indian Knowledge System	I. Generic IKS Course: basic knowledge of the IKS II. Subject Specific IKS Courses: advanced information pertaining to the subject: part of the major credit.	Subject Specific IKS related to Major

OJT	On-Job Training (Internship/Apprenticeship)	corresponding to the Major Subject	Related to the Major
FP	Field projects	corresponding to the Major Subject	Related to the Major
CC	Co-curricular Courses	Health and Wellness, Yoga education sports, and fitness, Cultural Activities, NSS/NCC and Fine/ Applied/Visual/ Performing Arts	Not Related to the Major and Minor
CE	Community Engagement and service		Not Related to the Major and Minor
RP	Research Project	corresponding to the Major Subject	Related to the Major

Programme Template:

Programme Degree		**Bachelor of Science (B. Sc.)**
Parenthesis if any (Specialization) e.g. History, Human Development, English, etc.		**Computer Science (CS)**
Preamble (Brief Introduction to the programme)		The Bachelor of Computer Science program at S.N.D.T. Women's University is designed in accordance with the National Education Policy 2020, aiming to cultivate a diverse set of skills and knowledge in students to meet the evolving demands of the digital age. Grounded in a holistic approach to education, this program not only fosters technical expertise but also emphasizes interdisciplinary learning, critical thinking, and ethical considerations. The curriculum is structured to provide a comprehensive foundation in computer science principles, practical problem-solving abilities, and proficiency in relevant tools and technologies. Students will embark on a journey of exploration and innovation, guided by a faculty committed to academic excellence and pedagogical innovation. **Key Highlights:** 1. **Foundation in Computer Science:** The curriculum begins with core courses such as Problem Solving using C, Fundamentals of Computer Science, and Object-Oriented Programming using C++, laying a solid groundwork in programming languages and computational thinking. 2. **Interdisciplinary Exposure:** In addition to core computer science courses, students will engage with diverse disciplines through Open Elective Courses, such as Web Technology, Multimedia Systems, and Environmental Science, fostering a well-rounded educational experience. 3. **Practical Application:** Practical skills are honed through courses like Office Automation Tools, Data Analytics using Spreadsheets, and Technical Writing, empowering students with hands-on experience and communication proficiency essential for professional success. 4. **Ethical and Social Responsibility:** Emphasis is placed on ethical considerations and social responsibility through courses like Professional Ethics, E-Waste Management, and Environmental Science, instilling a sense of accountability and sustainability in future technology professionals. 5. **Specialization and Electives**: Students have the opportunity to delve deeper into specialized areas of interest through elective courses such as Embedded Systems, Big Data Analytics, and Introduction to Artificial Intelligence, allowing for personalized learning paths aligned with individual career aspirations. 6. **Real-World Experience:** The program

		culminates in Field Projects, Internships, and On-the-Job Training, providing students with invaluable industry exposure and practical application of theoretical knowledge in real-world settings.
		Through a blend of rigorous academics, experiential learning, and holistic development, the Bachelor of Computer Science program at S.N.D.T. Women's University aims to nurture well-rounded professionals equipped to tackle the challenges of the digital era with creativity, integrity, and innovation
Programme Specific Outcomes (PSOs)		After completing this programme, Learner will
	1.	**Proficiency in Programming:** Graduates will be proficient in multiple programming languages, including C, C++, Python, and Java, enabling them to develop efficient and scalable software solutions to address diverse computing challenges.
	2.	**Problem-Solving Skills:** Graduates will possess strong analytical and problem-solving skills, capable of identifying, formulating, and solving complex computational problems using algorithmic techniques and data structures.
	3.	**Software Development Competence**: Graduates will demonstrate competence in software development methodologies and practices, including object-oriented design principles, version control, and software testing, ensuring the production of high-quality and reliable software systems.
	4.	**Technical Expertise:** Graduates will have a deep understanding of core computer science concepts and technologies, including operating systems, database management systems, and digital electronic and logic design, empowering them to adapt to emerging trends and technologies in the field.
	5.	**Interdisciplinary Knowledge:** Graduates will possess interdisciplinary knowledge in areas such as web technology, multimedia systems, and data analytics, allowing them to apply computer science principles in diverse domains and collaborate effectively in multidisciplinary teams.
	6.	**Communication and Presentation Skills:** Graduates will demonstrate effective communication and presentation skills, both orally and in writing, enabling them to articulate technical concepts, collaborate with stakeholders, and deliver compelling project reports and presentations.
	7.	**Ethical and Social Responsibility**: Graduates will exhibit ethical awareness and social responsibility in their professional practice, adhering to ethical codes of conduct, promoting inclusivity and diversity, and considering the societal implications of technological innovations.
	8.	**Professional Development:** Graduates will engage in lifelong learning and professional development activities, demonstrating the ability to adapt to evolving industry standards, acquire new skills, and pursue advanced studies or certifications as needed to enhance their career prospects.

	9.	**Industry Readiness:** Graduates will be equipped with the knowledge, skills, and practical experience necessary to transition seamlessly into the workforce, contributing effectively to industry projects, adapting to dynamic work environments, and demonstrating a commitment to continuous improvement and innovation.
	10.	**Leadership and Entrepreneurial Skills:** Graduates will demonstrate leadership qualities and entrepreneurial mindset, capable of initiating and managing projects, fostering innovation, and contributing positively to the growth and development of organizations or entrepreneurial ventures in the technology sector.
Eligibility Criteria for Programme		A Women candidate 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 in Science stream with 45% marks (40% for candidates belonging to Reserved category) with the following subjects :- • English • Any one of the Modern Indian Languages or Modern Foreign Languages or any classical Language or Information Technology/ Computer Science /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 recognized Board or Body Recognized as equivalent. Students who have not done mathematics at 12th Std. are needed to take a bridge course in mathematics and pass in university conducted test before semester 1 examination. OR Must have passed any three year Government recognized Diploma programme in Second Class.
Intake (For SNDT WU Departments and Conducted Colleges)		60

- *External Examination does not always mean Theory paper. It may practical examination, Product submission, projects, etc. checked by external examiners.*

- *Internal evaluation should not be Written Theory papers like Unit tests.*
  Internal marks will be acquired through practical, small group or individual Projects, activities, presentations, seminars, workshops, products, assignments, application-based work, reports, etc.

- Practical may be part of the main courses along with theory modules instead of having separate courses of practical work.

Structure with Course Titles

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

SN	Courses	Type of Course	Credits	Marks	Int	Ext
	Semester I					
1.1	Problem Solving Using C	Major (Core)	4 (2T+2P)	100	50	50
1.2	Fundamentals of Computer Science	Major (Core)	2	50	0	50
1.3	Open Elective Course-I	OEC	4	100	50	50
1.4	Office Automation Tools	VSC	2	50	50	0
1.5	SWAYAM/CHETNA/MOOCs	SEC	2	50	50	0
1.6		AEC	2	50	0	50
1.7		IKS	2	50	0	50
1.8		VEC	2	50	0	50
1.9	Co-curricular Course-I	CC	2	50	50	0
			22	**550**	**300**	**250**
	*1.5. SWAYAM/CHETNA/MOOCs subjects should be from Skill Enhancement course (SEC) category. *** 1.6,1.7,1.8,1.9 Co-Curricular Course (Health & Wellness, Yoga education, sports & fitness, Cultural activities, NSS, NCC and Fine/applied/visual/performing arts) will be provide by the University**						
	Semester II					
2.1	Object Oriented Programming using C++	Major (Core)	4 (2T+2P)	100	50	50
2.2	Operating System	Major (Core)	2	50	0	50
2.3	Introduction to Computer Hardware	Minor Stream	2	50	0	50
2.4	Open Elective Course-II	OEC	4	100	50	50
2.5	Data Analytics using Spreadsheet	VSC	2	50	0	50
2.6	Multimedia System	SEC	2	50	50	0
2.7		AEC	2	50	50	0
2.8		VEC	2	50	0	50
2.9	Co-curricular Course-II	CC	2	50	50	0
			22	**550**	**250**	**300**
	* 2.7,2.8 ,2.9 Co-Curricular Course (Health & Wellness, Yoga education, sports & fitness, Cultural activities, NSS, NCC and Fine/applied/visual/performing arts) will be provide by the University						

Exit with UG Certificate in Computer Science with 10 extra credits (44 + 10 credits)

SN	Courses	Type of Course	Credits	Marks	Int	Ext
	Semester III					
3.1	Data Structure	Major (Core)	4 (2T+2P)	100	50	50
3.2	Python Programming	Major (Core)	4 (2T+2P)	100	50	50
3.3	Digital Electronic and Logic Design	Minor Stream	4	100	50	50
3.4	Open Elective Course-III	OEC	2	50	0	50
3.5	Web Technology	VSC	2	50	50	0
3.6		AEC	2	50	0	50
3.7	Field Project	FP	2	50	50	0
3.8	Co-curricular course-III	CC	2	50	50	0
			22	**550**	**300**	**250**
	*3.6 and 3.8 will be provide by the University ***3.7 Field Project will be internal projects assigned to individual student on major subjects.**					
	Semester IV					
4.1	Design and Analysis of Algorithm	Major (Core)	4 (2T+2P)	100	50	50
4.2	Java Programming	Major (Core)	4 (2T+2P)	100	50	50
4.3	Introduction to Microprocessor and Microcontroller	Minor Stream	4	100	50	50
4.4	Open Elective Course-IV	OEC	2	50	0	50
4.5	SWAYAM/CHETNA/MOOCs	SEC	2	50	0	50
4.6		AEC	2	50	0	50
4.7	E- Waste Management	CEP	2	50	50	0
4.8	Co-curricular course-IV	CC	2	50	50	0
			22	**550**	**250**	**300**
	*4.6 and 4.8 will be provide by the University					

Open Elective Course

Open Elective Course -I	Digital Marketing
Open Elective Course -II	Intellectual Property Rights
Open Elective Course -III	Cyber Security
Open Elective Course -IV	Digital Forensic

Exit with UG Diploma with 10 extra credits (44 + 10 credits)

SN	Courses	Type of Course	Credits	Marks	Int	Ext
	Semester V					
5.1	Database Management System	Major (Core)	4 (2T+2P)	100	50	50
5.2	Advance Java Programming	Major (Core)	4 (2T+2P)	100	50	50
5.3	Software Engineering	Major (Core)	2	50	0	50
5.4	Elective – I	Major (Elective)	4	100	50	50
5.5	Embedded System	Minor Stream	4	100	50	50
5.6	Mobile Application Development using Android	VSC	2	50	50	0
5.7	Internship	FP	2	50	50	0
			22	**550**	**300**	**250**
	Semester VI					
6.1	Advance Database Management System	Major (Core)	4 (2T+2P)	100	50	50
6.2	Computer Network	Major (Core)	4	100	50	50
6.3	Software Testing	Major (Core)	2	50	0	50
6.4	Elective – II	Major (Elective)	4	100	50	50
6.5	Introduction to Digital Image Processing	Minor Stream	4	100	50	50
6.6	Internship	OJT	4	100	50	50
			22	**550**	**250**	**300**

Exit with Degree (3-year)

4-Year Degree with Honors

SN	Courses	Type of Course	Credits	Marks	Int	Ext
	Semester VII					
7H.1	Data Communication and Networking	Major (Core)	4	100	50	50
7H.2	Data Warehousing and Mining	Major (Core)	4	100	50	50
7H.3	Distributed System	Major (Core)	4	100	50	50
7H.4	Introduction to R programming	Major (Core)	2	50	50	0
7H.5	Elective-III	Major (Elective)	4	100	50	50
7H.6	Research Methodology	Minor Stream (RM)	4	100	50	50
			22	**550**	**300**	**250**
	Semester VIII					
8H.1	Introduction to Artificial Intelligence	Major (Core)	4	100	50	50
8H.2	Big Data Analytics	Major (Core)	4	100	50	50
8H.3	E-Commerce	Major (Core)	4	100	50	50
8H.4	Introduction to Machine Learning	Major (Core)	2	50	0	50
8H.5	Elective-IV	Major (Elective)	4	100	50	50
8H.6	On Job Training	OJT	4	100	50	50
			22	**550**	**250**	**300**

4-Year Degree with Research

SN	Courses	Type of Course	Credits	Marks	Int	Ext
	Semester VII					
7R.1		Major (Core)	4	100	50	50
7R.2		Major (Core)	4	100	50	50
7R.3		Major (Core)	2	50	0	50
7R.4		Major (Elective)	4	100	50	50
7R.5		Minor Stream (RM)	4	100	50	50
7R.6		Research Project	4	100	100	0
			22	**550**	**300**	**250**
	Semester VIII					
8R.1		Major (Core)	4	100	50	50
8R.2		Major (Core)	4	100	50	50
8R.3		Major (Core)	2	50	0	50
8R.4		Major (Elective)	4	100	50	50
8R.5		Research Project	8	100	100	100
			22	**550**	**250**	**300**

Electives

Elective-I	Elective-II
- Mobile Computing - Introduction to Data Science - Bioinformatics - Advance Java Programming	- Data Wrangling using Python - Internet of Things - Cloud Computing - ERP
Elective-III	**Elective-IV**
- Computer Vision - Introduction to Natural Language Processing - Project Management - Web 3.0	- Network Security - Introduction to Deep learning - Introduction to Metaverse - High Performance Computing

Course Syllabus

Semester I

1.1   Major (Core)

Course Title	**PROBLEM SOLVING USING C**
Course Credits	**2**
Course Outcomes	**After Completion of this Course, students will be able to**
	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. Design and develop applications using basic programming constructs, facilitating the transition to other languages.
Module1 (Credit1)	
Learning Outcomes	**After learning this 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.
Content Outline	• **Introduction to problem solving**: Concept: Steps in problem solving - (Define Problem, Analyze Problem, Explore Solution), Problem solving techniques - (Trial & Error, Brain Storming, 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.LanguageFundamentals:Characterset,CTokens,Keywords, Identifiers,Variables, Constant, Data Types, Comments

	- **Operators**: Types of operators, Precedence and Associativity, Expression. Statement and types of statements, Built-in Operators and function. Console based I/O and related built in I/O Function: printf(), scanf(), getch(), getchar(), putchar(), etc; Concept of header files, Preprocess ordirectives: #include,#define, Conditional statements and Loops
Module2 (Credit1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Gain proficiency in writing C programs to solve various problems.
	2. Learn the syntax and semantics of the C language, including its specific features such as pointers and memory management.
	3. Analyze the difference between structure and union.
	4. Design and handle operations of the files.

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

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

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.

References:-

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.

1.1 Major(Core)

Course Title	**PROBLEM SOLVING USING C (LAB)**
Course Credits	2
Course Outcomes	**After completion of this Course, the students will be able to**
	1. Apply algorithms by writing C code.
	2. Analyze and trace the execution of C programs.
	3. Evaluate the use of pointers, arrays, and the pre-processor in programs.
	4. Design programs that utilize derived data types and implement simple file operations.
Module1 (Credit 1)	
Learning Outcomes	**After learning this Module, learners will be able to**
	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(); • **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 2 (Credit 1)

Learning Outcomes	**After learning this Module, learners will be able to**
	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 multidimension 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

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

• **Module 1**

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.
4. Make use of header files and preprocessor directives to organize your code and define

any necessary constants.

- **Module 2**

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.

References:

1. Balagurusamy, E. (1990). C programming. Tata McGraw Hill.
2. Schildt, H. (2000). C: The Complete Reference (Fourth Edition). Tata McGraw-Hill Education Pvt. Ltd.
3. Ramkumar & Agrawal. (1996). Programming in ANSI C. Tata McGraw Hill.
4. Kanetkar, Y. P. (2008). Let Us C. Infinity Science Press.

1.2.Major (Core)

Course Title	**FUNDAMENTALS OF COMPUTER SCIENCE**
Course Credits	**2**
Course Outcomes	**After completion of this Course, the students will be able to**
	1. Apply computer fundamentals comprehensively.
	2. Analyze and solve problems using programming and software development skills.
	3. Evaluate proficiency in programming languages, software development life cycles, and debugging techniques.
	4. Design software with strong internet literacy and effective problem-solving skills.
Module1 (Credit1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Apply knowledge of essential computer components, including CPU, input/output devices, memory, and storage.
	2. Analyze internal data representation, including number systems, binary arithmetic, Boolean algebra, and logic gates.
	3. Evaluate the functions of various input/output devices and interfaces within computer systems.
	4. Design an understanding of basic computer organization and the roles of CPU, memory, and secondary storage.
Content Outline	• **Knowing computer:** What is Computer, Basic Applications of Computer; Components of Computer System, Central Processing Unit (CPU), Input/output Devices, Computer Memory, Concepts of Hardware and Software; Concept of Computing, Data and Information, classification of computers, various generations of computers; What is an Operating System; Different Popular Operating Systems; The User Interface, System Software: System software Vs. Application Software, Types of System Software, Introduction and Types of Operating Systems
	• **Internal data Representation in Computers and Digital System Design:** Number Systems Used in Computers, Converting from One Number System to Another, Binary Arithmetic, Boolean Algebra, Boolean Functions, Logic Gates, Logic Circuits
	• **Computer Architecture:** Basic Functions of a Computer, Basic Computer Organization, CPU Architectures, Memory Architectures, Secondary Storage, Classification of Secondary Storage, Memory Storage Devices, Input- Output Devices ,Input Devices, Output Devices, I/O Interfaces.
Module2 (Credit1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Apply knowledge of computer language evolution and types.
	2. Analyze programming constructs, algorithms, and flowcharting skills.
	3. Evaluate the roles of language processors, software development, and the SDLC.
	4. Design an understanding of basic computer networks, internet connectivity, and the architecture of the World Wide Web.

Content Outline	• **Computer Languages:** Definition, Generations of computer languages, Types of Languages, Language Processors: Assembler, Interpreter, Compiler, Linker and Loader. Programming constructs, Algorithm & flowchart. • **Software:** Basic Concepts and Terminologies, What is Software? ,Relationship between Hardware and Software, Software Development Life Cycle (SDLC), Advantages of SDLC Model, Limitations of SDLC Model, Software Testing and Debugging • **Introduction to Internet:** WWW and Web Browsers: Basic of Computer networks; LAN, WAN; Concept of Internet; Applications of Internet; connecting to internet; What is ISP; Knowing the Internet; Basics of internet connectivity related troubleshooting, World Wide Web; Web Browsing software, Search Engines; Understanding URL; Domain name; IP Address; Using e-governance website

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

• **Module 1** :

Write an essay discussing the fundamental components and concepts of computer systems.

1. Explain the basic components of a computer system, including CPU, input/output devices, memory, and storage.
2. Describe the internal data representation in computers, covering number systems, binary arithmetic, Boolean algebra, and logic gates.
3. Discuss computer architecture, including CPU architectures, memory architectures, and secondary storage.
4. Analyze the functions and roles of input/output devices and interfaces within computer systems.
5. Provide examples and illustrations where necessary to clarify concepts.

• **Module 2** :

Develop a presentation on computer languages, software development, and the basics of internet connectivity.

1. Explain the evolution and types of computer languages, including the generations of languages and the role of language processors (assembler, interpreter, compiler, linker, and loader).
2. Discuss programming constructs, algorithms, and flowcharting skills.
3. Describe basic concepts and terminologies related to software, including the software development life cycle (SDLC), software testing, and debugging.
4. Provide an introduction to the internet, covering basic computer network concepts (LAN, WAN), internet connectivity, and the architecture of the World Wide Web.
5. Include examples and case studies to illustrate key points effectively.

References: -

1. Sinha, P. K., & Sinha, P. (2004). Computer Fundamentals. BPB Publications.
2. Srivastava, C. (2010). Fundamentals of Information Technology. Kalyani Publishers.
3. Rajaraman, V. (2003). Fundamentals of Computers (4th ed.). PHI Publication.
4. Jain, R. K. (2010). Fundamentals of Programming. S. Chand Publication.

Semester I

1.3. OEC (Open Elective Course -I)

Course Title	Digital Marketing
Course Credits	**4**
Course Outcomes	**After completion of this Course, the students will be able to**
	1. Apply principles and foundations of digital marketing.
	2. Analyze data to optimize marketing strategies and campaigns.
	3. Evaluate digital marketing performance using analytics tools.
	4. Design effective marketing using digital channels, customer insights, and segmentation.
Module1 (Credit1)	
Learning Outcomes	**After learning this module, learners will be able to**
	1. Apply principles and techniques of digital marketing using various channels and tools.
	2. Analyze and understand SEO techniques.
	3. Evaluate the effectiveness of digital marketing strategies.
	4. Design comprehensive digital marketing campaigns.

Content Outline	• **Basics of Websites & Digital Marketing: -**
	   ➢ Fundamental of Digital Marketing: Concept, Scope, Areas to Explore
	   ➢ Building Websites on Weebly, WordPress &Blogger
	   ➢ Designing: Canva Tool for Image Editing and Photoshop Graphics
	
	 • **SEO -On Page Optimization: -**
	   ➢ Broken links, Backlinks, W3 Errors, Keyword research &optimization Heading Tag Optimization: Reporting, Suggestion and Implementing Backlinks, Titles & Meta Descriptions, Website Content Optimization Meta& Title Tags Adjustment, XLM Site Map Setup, Robot.txt Validation Google Analytics and Webmaster Tool Setup
	   ➢ SERP - rankings on google, Plugins Installation and Monitoring Heading Tag Optimization: Reporting, Suggestion and Implementing Duplicate Content Reporting
	   ➢ Duplicate Content Rewording/rewriting using seo target keywords
	   ➢ Plugins &Internal Linking, Permalinks Optimization: Reporting, Suggestion and Implementing

Module2 (Credit1)	
Learning Outcomes	**After learning this module, learners will be able to**
	1. Apply strategies for effective campaigns.
	2. Analyze website metrics using Google Analytics.
	3. Evaluate important metrics for SEM campaigns.
	4. Design and optimize campaigns based on analytics data.
Content Outline	• **Social Media Optimization:** - Custom Graphics and Setting Profile with about/ hours and other information of business: For your Social Media Account
Module3 (Credit1)	
Learning Outcomes	**After learning this module, learners will be able to**
	1. Apply business analytics techniques for data-driven marketing decisions.
	2. Analyze data insights to optimize marketing strategies and campaigns.
	3. Evaluate the effectiveness of data-driven marketing decisions.
	4. Design marketing strategies based on interpreted data insights.
Content Outline	• **Social Media Content Creation and Posting**: 5 Social Media Platform: 2 posts for each platform and1ImagePost and1interesting post found related to the business you are in on web and shared on your page each day. Along with: Commenting, Follows, Likes, Shares • **Paid Advertising: Email Marketing, FB Marketing & Google AdWords** Facebook Marketing: Ad Plan, Ad Setup with Banner Images, Monitoring & Reporting (1 Ad: Website Conversion, Apps Installation, Promote Page Etc.) Google AdWords: Search Ads Type: Keyword Research, 1Campaign, 2Adgroups, 5Ads.DisplayAds: 2Banner Graphics keyword Research, 1AdGroup, 2 Ads. • **Email Marketing**: Content Writing as per Offers, services & Discounts Custom Template Building, HTML Conversion, Use of Emailing Software: Account Creation, Creating Customer List, Sending Emails
Module4 (Credit1)	
Learning Outcomes	**After learning this module, learners will be able to**
	1. Apply appropriate metrics and analytics tools to measure digital marketing campaign performance.
	2. Analyze data gathered from campaigns to evaluate their effectiveness.
	3. Evaluate the performance of digital marketing campaigns based on measured metrics.
	4. Design strategies for optimizing digital marketing campaign performance based on evaluation results.
Content Outcomes	• **Video sharing platform Optimization**: Posting, pinning, repining Analytics: Search engine Analytics, Installing Analytics, How to Study search engine Analytics, Interpreting Bars & Figures, How Analytics can Help SEO, Advanced Reporting, Webmaster Central, Bing/Yahoo, Open Site Explorer, Website Analysis using various SEO Tools available. **Reporting and Monitoring: -** SEO REPORT/plan, Initial website ranking and evaluation report Analytics report and web master report Ad Words Report, Fb Insight, Marketing Media Audit Reports ERP Report, Competitor

	Analysis Report(Media Audit Report)

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

- **Module 1**:

 Design and execute an SEO audit for a website.

1. Perform on-page optimization tasks such as checking for broken links, W3 errors, keyword research, optimizing heading tags, titles, and meta descriptions.

2. Implement necessary changes based on the audit findings, including fixing broken links, optimizing content, and adjusting meta tags.

3. Set up essential tools like Google Analytics and Google Webmaster Tools for monitoring website performance and tracking SEO metrics.

4. Create a comprehensive report summarizing the audit findings, implemented changes, and recommendations for further optimization.

- **Module 3**:

 Develop a social media marketing campaign for a business.

1. Create custom graphics and optimize social media profiles for the selected business on major platforms like Facebook, Twitter, Instagram, etc.

2. Plan and execute a paid advertising campaign using platforms like Google AdWords and social media ads.

3. Design and send email marketing campaigns, including content creation, template building, and email list management.

4. Monitor and analyze the performance of the campaigns using relevant metrics and analytics tools.

5. Prepare detailed reports showcasing campaign performance, including insights, key metrics, and recommendations for improvement.

References

1. Chaffey, D., & Ellis-Chadwick, F. (2019). Digital Marketing: Strategy, Implementation and Practice (7th ed.). Pearson Education Limited.
2. Ryan, D. (2017). Understanding Digital Marketing: Marketing Strategies for Engaging the Digital Generation (4th ed.). Kogan Page.
3. Kotler, P., Kartajaya, H., & Setiawan, I. (2017). Marketing 4.0: Moving from Traditional to Digital. Wiley.

Semester I

1.4. VSC

Course Title	Office Automation Tools
Course Credits	2
Course Outcomes	**After going through the course, 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
	• **Write**r -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 Slide Shows, Setting Up and Customizing Impress

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

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.

Semester I

1.5. SEC (SWAYAM/CHETNA/MOOCs)

Semester I

1.9. CC (Co-Curriculum Course-I)

Semester II

2.1 Major (Core)

Course Title	Object Oriented Programming Using C++
Course Credits	2
Course Outcomes	**After going through the course, learners will be able to**
	1. Apply object-oriented programming concepts in C++.
	2. Analyze problems and develop C++ programs to solve them.
	3. Evaluate the use of file input/output in C++.
	4. Design solutions using object-oriented programming principles in C++.
Module 1(Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Apply fundamental programming concepts including variables, data types, control structures, functions, arrays, and objects.
	2. Analyze and implement object-oriented programming concepts like objects, classes, and defining functions and variables.
	3. Evaluate the use of object-oriented programming in solving programming problems.
	4. 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**
	1. Apply concepts of constructors, inheritance (including its types), and polymorphism in C++.
	2. Analyze file input/output handling in C++ and class templates.
	3. Evaluate the effectiveness of constructors, inheritance, and polymorphism in solving programming tasks.
	4. 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 Destructors 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.

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

Module 1:

- Develop a console-based application in C++ that demonstrates the implementation of fundamental programming concepts and object-oriented principles.
- Create a C++ program that includes classes representing real-world entities (e.g., a student, a car, a bank account).
- Implement basic functionality within these classes, such as setting and retrieving object attributes, defining member functions for data manipulation, and demonstrating encapsulation and abstraction.
- Utilize inheritance to create derived classes that inherit properties and behaviors from base classes, showcasing the concept of code reusability.
- Incorporate polymorphism by defining virtual functions and overriding them in derived classes to demonstrate runtime polymorphism.
- Implement file input/output operations to store and retrieve data related to objects, showcasing the handling of persistent data using C++.
- Document your code thoroughly and provide comments to explain the purpose and functionality of each component.

Module 2 :

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

References:

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

2.1 Major (Core)

Course Title	Object Oriented Programming Using C++ (Lab)
Course Credits	2
Course Outcomes	After going through the course, learners will be able to
	1. Apply object-oriented programming concepts by creating programs with classes and objects in C++.
	2 Analyze the implementation of object-oriented programming concepts in C++.
	3 Evaluate the effectiveness of stream I/O and file I/O in developing applications.
	4 Design object-oriented programs using templates and exceptional handling techniques in C++.
Module 1(Credit 1)	
Learning Outcomes	After learning the module, learners will be able to
	1. Apply the learned concepts by writing simple programs using classes and objects in C++.
	2. Analyze the implementation of object-oriented programming concepts in C++ to understand their functionality and usage.
	3. Evaluate the effectiveness of object-oriented programming in solving programming tasks.
	4. Design solutions using object-oriented programming concepts to improve code structure and modularity in C++.
Content Outline	• **Simple Programs** on fundamental Data Types and I/O operators, Derived data types, Symbolic constants, variables and Reference variables
	• **Operators and decision control structures:** Programs to implement if statements, Switch statements, Loop statements
	• **Functions in C++:** Programs based on feature of an inline function, default arguments, function overloading, call by value, call by address, Call by reference.
	• **Advanced Language Constructs:** Program based on concept of Linear array, multidimensional array, pointer
Module 2(Credit 1)	
Learning Outcomes	After learning the module, learners will be able to
	1. Apply the concepts of inheritance, constructors, and operator overloading by writing simple programs in C++.
	2. Analyze the process of reading and creating text files using C++ programs.

	3. Evaluate the effectiveness of inheritance, constructors, and operator overloading in solving programming tasks.
	4. Design programs that utilize inheritance, constructors, and operator overloading for improved code organization and functionality in C++.
Content Outline	• **Object and Classes:** Program based on class, member function, One function calls another Function, arrays within a class, How to use a array in a class, static class member, Class member should be astatic variable, Array of object. • **Friend Function** • **Inheritance:** • **Constructor and Destructor**: • **Polymorphism:** • **Files and streams:** • **Class templates:** Implementations of Class template, Class template with multiple parameters, Function template

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

Module 1:

- Develop a set of C++ programs that demonstrate the fundamental concepts of object-oriented programming (OOP).
- Write a C++ program that utilizes classes and objects to represent real-world entities (e.g., a car, a student, a bank account).
- Implement simple programs to demonstrate the usage of derived data types, symbolic constants, variables, and reference variables.
- Create programs that utilize decision control structures such as if statements, switch statements, and loop statements to demonstrate control flow in C++.
- Design and implement functions within classes to showcase encapsulation and abstraction principles.
- Provide comments and documentation within your code to explain the purpose and functionality of each component.

Module 2 :

- Design a series of C++ programs that utilize inheritance, constructors, operator overloading, and file handling.
- Implement a program that demonstrates the concept of inheritance by creating a base class and derived classes that inherit from it, showcasing the reuse of code and the specialization of functionality.
- Write programs that utilize constructors and destructors to initialize and clean up resources within objects, demonstrating the importance of proper resource management.
- Implement operator overloading in C++ programs to demonstrate the ability to define custom behavior for operators such as +, -, *, etc.
- Develop programs that read from and write to text files using file handling techniques in C++, showcasing the ability to manipulate external data.
- Utilize class templates to create generic data structures or algorithms that can operate on different data types, demonstrating the power of template-based programming.
- Test your programs with various scenarios and inputs to ensure correctness and functionality.

- Provide comprehensive documentation explaining the design decisions, implementation details, and usage instructions for each program.

References:-

1 Balguruswamy, E. (2008). Object Oriented Programming with C++. Tata McGraw–Hill Education.
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.

2.2. Major (Core)

Course Title	Operating System
Course Credits	2
Course Outcomes	**After going through the course, learners will be able to**
	1. Apply knowledge of computer system organization and architecture to solve complex problems.
	2. Analyze the structure, operations, and design principles of operating systems.
	3. Evaluate virtualization techniques, including virtual machines and operating system generation, for efficiency and effectiveness.
	4. Design processor management and scheduling strategies for multi-processor systems to optimize performance.
Module 1(Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Apply knowledge of computer system organization and architecture, with a focus on the role of operating systems.
	2. Analyze the structure and operations of operating systems, including key components like process management, memory management, storage management, protection, and security.
	3. Evaluate the various operating system services available to users and applications for efficiency and effectiveness.
	4. Design understanding of the user-operating system interface, system calls, and types of system programs to enhance system functionality and usability.
Content Outline	• **Introduction to Operating Systems (OS):** Computer-System Organization, Computer-System Architecture, Operating-System Structure, Operating-System Operations, Process Management, Memory Management, Storage Management, Protection and Security, Distributed Systems, Special-Purpose Systems, Computing Environments.
	• Operating-System Services, User Operating-System Interface, System Calls, Types of System Calls, System Programs, Operating-System Design and Implementation, Operating-System Structure, Virtual Machines, Operating-System Generation.
	• **Processor Management:** Process concept, Process scheduling, Operations on Processes, Interprocess Communication, Multithreading models, threading issues, Process scheduling algorithms, Thread scheduling, Multiple processor Scheduling.
	• **Process Coordination**: Synchronization, Semaphores, Monitors, Deadlocks characterization, Methods for handling deadlocks, Deadlock prevention, Deadlock Avoidance, Deadlock detection, recovery from deadlock.
Module 2(Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Apply various memory management techniques such as swapping, contiguous memory allocation, paging, and segmentation in practical scenarios.
	2. Analyze the structure and function of the page table to understand its role in memory addressing.
	3. Evaluate fundamental file concepts, including file access methods, directory structures, file sharing, and protection

	mechanisms, to ensure efficient file management.
	4. Design efficient file systems by understanding their organization, structure, directory implementation, allocation methods, and free-space management to optimize storage and retrieval operations.
Content Outline	• **Memory Management**: Swapping, Contiguous Memory Allocation, Paging, Structure of the Page Table, Segmentation • **Virtual memory Management:** Demand Paging, Copy- on- Write, Page replacement, Allocation of Frames, Thrashing. • **File Management:** File Concept, File Access Methods, Directory Structure, File Sharing, File Protection, File-System Structure, File- System Implementation, Directory Implementation, Allocation Methods, Free-Space Management, Efficiency and Performance, Recovery, Log-Structured File Systems, NFS.

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

Module 1:

- Design a presentation that explains the fundamental concepts of operating systems and their key components.
- Create slides that cover topics such as computer system organization, architecture, operating system structure, and operations.
- Explain the role of operating systems in process management, memory management, storage management, protection, and security.
- Describe the various operating system services available to users and applications, highlighting their efficiency and effectiveness.
- Discuss the user-operating system interface, system calls, and types of system programs to enhance system functionality and usability.
- Include examples and illustrations to clarify complex concepts and engage the audience effectively.

Module 2:

- Develop a research paper that explores memory management techniques and file system concepts in operating systems.
- Discuss various memory management techniques such as swapping, contiguous memory allocation, paging, and segmentation, explaining their advantages, disadvantages, and practical applications.
- Analyze the structure and function of the page table in virtual memory management, highlighting its role in memory addressing and address translation.
- Evaluate fundamental file concepts, including file access methods, directory structures, file sharing, and protection mechanisms, to ensure efficient file management.
- Design efficient file systems by understanding their organization, structure, directory implementation, allocation methods, and free-space management to optimize storage and retrieval operations.
- Provide real-world examples and case studies to illustrate the implementation and impact of memory management and file system concepts in operating systems.
- Include references to relevant literature and research papers to support your analysis and findings.

References:

1. Silberschatz, A., Galvin, P. B., & Gagne, G. (2005). Operating System Concepts (7th ed.). John Wiley & Sons, Inc.
2. Milenkovic, M. (2001). Operating Systems Concepts And Design (2nd ed.). McGraw-Hill International Editions.
3. Stallings, W. (2005). Operating Systems: Internals and Design Principles (5th ed.). Prentice Hall.
4. Tanenbaum, A. (2007). Modern Operating Systems (3rd ed.). Pearson Education.

Semester II

2.3. Minor Stream

Course Title	Introduction To Computer Hardware
Course Credits	**2**
Course Outcomes	**After going through the course, learners will be able to**
	1. Apply knowledge of computer hardware components to diagnose and solve hardware problems effectively.
	2. Analyse differences between hardware and software, evaluating their roles in computing systems.
	3. Evaluate network types (LAN, MAN, WAN) for advantages and disadvantages in various environments.
	4. Design PC assembly strategies and troubleshooting methodologies to ensure system reliability and performance.
Module 1(Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Apply knowledge of computer hardware components to propose effective hardware upgrade solutions based on system requirements.
	2. Analyse the characteristics and functionalities of different types of computer memory to optimize performance in computing tasks.
	3. Evaluate the advantages and disadvantages of LAN, MAN, and WAN to design appropriate network configurations for specific organizational needs.
	4. Design strategies for diagnosing and resolving computer hardware issues using systematic diagnostic methods, ensuring efficient problem-solving.
Content Outline	• **Fundamentals of Computer Hardware**: What is Computer Hardware and Hardware Upgrade, Computer Hardware Parts(Components), Hardware vs Software, Hardware Virtualization, Hardware as a service, Computer Hardware Problems and Diagnostic Methods • **Computer Memory:** Definition, Characteristics of Main Memory, how does Computer Memory work, Types of Computer Memory, Register Memory, Cache Memory, Primary and Secondary Memory and its types, RAM, ROM and Memory units • **Computer Network:** Basics of Network, LAN, MAN and WAN along with advantages and disadvantages.
Module 2(Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Apply knowledge of PC assembly by selecting appropriate components, adhering to safety protocols, and executing step-by-step assembly procedures.
	2. Analyse the advantages of different types of computer cables and processors to optimize system performance and connectivity.
	3. Evaluate troubleshooting approaches to diagnose and resolve hardware and software issues in microcomputers and peripheral

	equipment.
	4. Design effective strategies for managing faulty components, installing I/O devices, and ensuring system stability during boot processes.
Content Outline	• **PC Assembling and Troubleshooting:** • **Assembling:** a) How to build a computer: Choosing the right components, Safety Measures, Steps to build a computer b) Types of Computer Cables and its advantages c) Types of Processors d) Working of Printers and Scanners and its types e) Microcomputers and Motherboards and its types and selection of right motherboard f) Drivers role and types • **Troubleshooting:** Diagnose and troubleshooting of microcomputer/computer system • **hardware & software and other peripheral equipment:** Approaches to solve a PC problem, troubleshooting a failed boot before the OS is loaded, Different approaches to installing and supporting I/O devices, Managing Faulty Components.

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

Module 1

- Application of theoretical knowledge to practical scenarios.
- Analysis and evaluation of hardware components, memory types, and network configurations.
- Design of effective strategies for hardware troubleshooting.
- Clarity, coherence, and depth of the written report and presentation.

Module 2

- Application of PC assembly knowledge to build a functional custom PC.
- Analysis of different types of computer cables and processors in relation to system performance.
- Evaluation of troubleshooting approaches and effectiveness in resolving hardware and software issues.
- Design of clear and effective strategies for managing faulty components and ensuring system stability.

References

1. Meyers, M. (2017). Introduction to PC Hardware and Troubleshooting. McGraw Hill Education.
2. Lotia, M. (2006). Modern Computer Hardware Course. BPB Publication.
3. Zacker, C., & Rourke, J. (2017). PC Hardware: The Complete Reference. McGraw Hill Education.

Semester II

2.4.   OEC (Open Elective Course-II)

Course Title	Intellectual Property Rights
Course Credits	4
Course Outcomes	**After going through the course, learners will be able to**
	1. Apply knowledge of Intellectual Property Rights (IPR) principles to effectively protect their creative work from infringement.
	2. Analyse the various uses and applications of Intellectual Property (IP) in different contexts such as business, innovation, and creative industries.
	3. Evaluate strategies for utilizing Intellectual Property to safeguard and promote their own creative works and innovations.
	4. Design a comprehensive understanding of the registration processes for Copyrights, Patents, and Trademarks, ensuring they can navigate and utilize these protections effectively.
Module1 (Credit1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Apply knowledge of the justifications and rationales for protecting intellectual property to real-world scenarios, such as explaining the economic benefits of IP protection in innovation-driven industries.
	2. Analyse the ethical and legal frameworks surrounding intellectual property protection, examining how different cultural and legal contexts influence these frameworks.
	3. Evaluate the impact of intellectual property protection on technological advancement, creativity, and societal development, considering both positive and negative aspects.
	4. Design strategies to advocate for and defend intellectual property rights based on a deep understanding of the philosophical, economic, and legal justifications for IP protection.
Content Outline	• **Basic Principles and Acquisition of Intellectual Property Rights:** Philosophical Aspects of Intellectual Property Laws, Basic Principles of Patent Law, Patent Application procedure, drafting of a Patent Specification, Understanding Copyright Law, Basic Principles of Trade Mark, Basic Principles of Design Rights, International Background of Intellectual Property.
Module2 (Credit1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Apply knowledge of different countries' Intellectual Property Rights (IPR) acts to navigate international business and legal contexts effectively.
	2. Analyse the similarities and differences between IPR acts across various countries, considering cultural, economic, and legal factors.
	3. Evaluate the effectiveness of different countries' IPR acts in protecting intellectual property rights, considering enforcement mechanisms and international agreements.
	4. Design strategies for multinational corporations or creators to navigate and comply with diverse IPR acts globally, ensuring comprehensive protection of intellectual property assets.

Content Outline	• **Information Technology Related Intellectual Property Rights**: **Computer Software and Intellectual Property-** Objective, Copyright Protection, Reproducing, Defences, Patent Protection.
	• **Database and Data Protection-**Objective, Need for Protection, UK Data Protection Act, 1998, US Safe Harbor Principle, Enforcement.
	• **Protection of Semi-conductor Chips-**Objectives, Justification of protection, Criteria, Subject matter of Protection, WIPO Treaty, TRIPs, SCPA.
	• **Domain Name Protection-** Objectives, domain name and Intellectual Property, Registration of domain names, disputes under intellectual Property Rights, Jurisdictional Issues, and International Perspective.

Module3 (Credit 1)

Learning Outcomes	After learning the module, learners will be able to
	1. Apply knowledge of different patents and copyrights information to effectively distinguish between types of intellectual property protections.
	2. Analyse the detailed process of patenting and development, including requirements, timelines, and implications for innovation and business.
	3. Evaluate the procedure of trademark development in terms of legal requirements, market impact, and brand protection strategies.
	4. Design strategies for creators and innovators to navigate and utilize patenting, copyrighting, and trademarking processes to safeguard their intellectual property effectively.
Content Outline	• **Patents (Ownership and Enforcement):** Objectives, Rights, Assignments, Defenses in case of Infringement.
	• **Copyright (Ownership and Enforcement): Copyright:** Objectives, Rights, Transfer of Copyright, work of employment Infringement, Defenses for infringement.
	• **Trademark (Ownership and Enforcement): Trademarks:** Objectives, Rights, Protection of goodwill, Infringement, Passing off, Defenses. Designs: Objectives, Rights, Assignments, Infringements, Defenses of Design Infringement.

Module4 (Credit1)

Learning Outcomes	After learning the module, learners will be able to
	1. Apply knowledge of new developments in Information Technology and Cyber laws to adapt and integrate emerging technologies responsibly.
	2. Analyse the basics of Cyber laws and new technologies, including their implications on privacy, security, and digital rights.
	3. Evaluate case studies related to Cyber laws and new technology to understand their practical applications and legal challenges.
	4. Design strategies for implementing Cyber laws effectively in response to technological advancements, ensuring legal compliance and ethical practices in information technology.
Content Outline	• **Enforcement of Intellectual Property Rights:** Civil Remedies, Criminal Remedies, Border Security measures.

	Practical Aspects of Licensing: Benefits, Determinative factors, important clauses, licensing clauses.
	• **Cyber Law:** Basic Concepts of Technology and Law: Understanding the Technology of Internet, Scope of Cyber Laws, Cyber Jurisprudence Law of Digital Contracts: The Essence of Digital Contracts, The System of Digital Signatures, The Role and Function of Certifying Authorities, The Science of Cryptography.
	• **Case studies:** Case studies related to different cyber crimes and punishment can be given.

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

Module 1:

- Apply knowledge of the justifications and rationales for protecting intellectual property to real-world scenarios in the technology sector.
- Analyse the ethical and legal frameworks surrounding intellectual property protection, considering global cultural and legal contexts.
- Design strategies to advocate for and defend intellectual property rights within the firm's innovation-driven initiatives.

Module 2:

- Evaluate the effectiveness of different countries' Intellectual Property Rights (IPR) acts in protecting intellectual property rights.
- Analyse the similarities and differences between IPR acts across various countries, considering cultural, economic, and legal factors.
- Design strategies for navigating and complying with diverse global IPR acts to ensure comprehensive protection of intellectual property assets.

Module 3:

- Apply knowledge of patents, copyrights, trademarks, and designs to develop a cohesive intellectual property protection strategy.
- Analyse the detailed processes of patenting, copyrighting, and trademark development, including legal requirements and implications.
- Evaluate case studies related to intellectual property disputes to understand practical applications and legal challenges.

Module 4:

- Design strategies for implementing Cyber laws and intellectual property enforcement measures to protect digital assets.
- Analyse the basics of Cyber laws and their implications on privacy, security, and digital rights in international contexts.
- Develop a plan for integrating new technologies while ensuring legal compliance and ethical practices in intellectual property management and cybersecurity.

Reference:

1. Sood, V. (2017). Cyberlaw. McGraw Hill Education.
2. Leland, C. R. (1995). Licensing Art & Design. Allworth Press.
3. Leland, C. R. (1995). A Professional's Guide to Licensing and Royalty Agreements. Allworth Press.
4. Warda, M. (2002). How To Register Your Own Copyright. Sphinx Publishing.

Semester II

2.5   VSC

Course Title	Data Analytics Using Spreadsheet
Course Credits	**2**
Course Outcomes	**After going through the course, learners will be able to**
	1. Apply comprehensive understanding of Microsoft Excel to efficiently navigate through toolbars, interface components, and commands for various tasks.
	2. Analyse spreadsheet management techniques, including manipulation of columns, rows, and ranges, and the application of advanced functions and formulas to organize and interpret data effectively.
	3. Evaluate advanced Excel techniques such as chart creation, data analysis, PivotTable utilization, and macro implementation for generating meaningful insights and automating tasks in professional and academic contexts.
	4. Design strategies for leveraging Excel's capabilities to streamline workflows, improve data accuracy, and enhance decision-making processes using advanced features and functionalities.
Module 1(Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Apply essential Excel features and interface components to efficiently navigate and utilize the software for data management tasks.
	2. Analyse data structures in Excel, including columns, rows, and ranges, to effectively organize, format, and manage datasets.
	3. Evaluate the application of basic and advanced Excel formulas and functions for accurate data analysis and computation in academic and professional contexts.
	4. Design strategies for utilizing Excel functionalities to enhance data manipulation, improve data accuracy, and optimize data presentation using formulas and functions.
Content Outline	• **Introduction to Excel** : About Excel & Microsoft, Uses of Excel, Excel software, Spreadsheet window pane, Title Bar, Menu Bar, Standard Toolbar, Formatting Toolbar, the Ribbon, File Tab and Backstage View, Formula Bar, Workbook Window, Status Bar, Task Pane, Workbook & sheets • **Columns & Rows**: Selecting Columns & Rows, Changing Column Width & Row Height, Autofitting Columns &Rows, Hiding/Unhiding Columns & Rows, Inserting & Deleting Columns & Rows, Cell, Address of a cell, Components of a cell – Format, value, formula, Use of paste and paste special • **Functionality Using Ranges:** Using Ranges, Selecting Ranges, Entering Information Into a Range, Using AutoFill, text split using text to column, remove duplicates. • **Creating Formulas:** Using Formulas, Formula Functions –

	Sum, Average, if, Count, max, min, Proper, Upper, Lower, Using AutoSum, • **Advance Formulas:** Concatenate, Vlookup, Hlookup, Index, Match, Countif, if, Sumif, Text, Trim, Conditional Formatting.
Module 2(Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	1. Apply proficiency in creating and formatting various types of charts in spreadsheets to visually represent data and insights effectively.
	2. Analyse data analysis techniques such as sorting, filtering, and data validation to manipulate and interpret spreadsheet data efficiently.
	3. Evaluate the competency in utilizing PivotTables for dynamic data summarization and analysis, enabling extraction of meaningful insights from large datasets.
	4. Design strategies for utilizing charting tools, data analysis techniques, and PivotTables to enhance data visualization, manipulation, and interpretation skills in professional and academic settings.
Content Outline	• **Spreadsheet Charts**: Creating Charts, Different types of charts, Formatting Chart Objects, Changing the Chart Type, Showing and Hiding the Legend, Showing and Hiding the Data Table • **Data Analysis:** Sorting, Filter, Text to Column, Data Validation • **Pivot Tables** : Creating PivotTables, Manipulating a PivotTable, Using the PivotTable Toolbar, Changing Data Field, Properties, Displaying a PivotChart, Setting PivotTable Options, Adding Subtotals to Pivot Tables • **Spreadsheet Tools:** Moving between Spreadsheets, Selecting Multiple Spreadsheets, Inserting and Deleting Spreadsheets Renaming Spreadsheets, Splitting the Screen, Freezing Panes, Copying and Pasting, Data between Spreadsheets, Hiding , Protecting worksheets • **Making Macros:** Recording Macros, Running Macros

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

Module 1:

- Assignment 1: Introduction to Excel: Create a presentation or infographic detailing
- Excel's overview, uses, software features, and interface elements.
- Assignment 2: Columns & Rows Mastery: Manipulate columns and rows, adjust widths and heights, hide/unhide, insert/delete, and understand cell addressing.

Module 2:

- Assignment 3: Formula Mastery: Apply basic formulas (e.g., sum, average, Max, Min) and advanced functions (e.g., VLOOKUP, COUNTIF, Index+Match) to analyze data. Analyze data using conditional formatting.

- Activity: Chart Creation and Analysis: Create various chart types, format objects,   change chart types, and analyze data visually for insights.
- Assignment 4: Data Analysis and PivotTables: Use sorting, filtering, text to columns, data validation, and PivotTables for dynamic data analysis and summarization.
- Activity: Spreadsheet Tools Exploration Practice moving between sheets, selecting multiple sheets, inserting/deleting/renaming sheets, and utilizing features like splitting, freezing, copying/pasting. Macro Making Task: Record, run, and delete macros to automate repetitive tasks in Excel.

References:

1. Walkenbach, J. (2019). Excel 2019 Bible. Wiley.
2. Frye, C. D. (2019). Excel 2019 All-in-One For Dummies. For Dummies.
3. McFedries, P. (2019). Excel 2019 Formulas and Functions. Wiley.
4. Alexander, M., & Kusleika, D. (2019). Excel Macros For Dummies. For Dummies.
5. DeMarco, T. (2017). The Excel Annoyances Book. O'Reilly Media.

Semester II

2.6 SEC

Course Title	**Multimedia System**
Course Credits	**2**
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. • **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 2(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	• **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
	• **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

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

Module 1:
Assignment: 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.

Assignment: Font Analysis and Design:

- Analyze different font types and their usage in multimedia.
- Design a multimedia presentation focusing on font selection and its impact on communication.

Assignment: 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.

Assignment: 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.

Assignment: 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.

Module 2:

Assignment: Image Editing and Web Production Task:

- Use image editing software to produce web-ready graphics, focusing on selection, layering, and optimization techniques.

Assignment: 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.

Assignment: 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.

Assignment: 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.

References

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.

Semester II

2.9. CC (Co-curriculum Course-II)