Bachelor of Computer Applications (BCA)

TentativeTemplate

Terminologies

Abbreviation	Full-form	Remarks	RelatedtoMajorand MinorCourses
Major(Core)	MainDiscipline		
Major(Elective)	ElectiveOptions		related to theMajorDiscipline
MinorStream	OtherDisciplines(Inter/ Multidisciplinary) notrelatedto theMajor	either from the sameFacultyorany otherfaculty	
OEC	OpenElectiveCourses/ Generic		NotRelatedto theMajorand Minor
VSEC	Vocational and SkillEnhancementCourses		
VSC	VocationalSkillCourses		NotRelatedto theMajorand Minor
SEC	SkillEnhancementCourses		NotRelatedto theMajorand Minor
AEC	AbilityEnhancement Courses	Communication skills,critical reading, academicwriting,etc.	Not Related tothe Major andMinor
VEC	ValueEducationCourses	Understanding India,Environmentalscience/education, Digitalandtechnologicalsolutions,Health & Wellness, Yogaeducation, sports, andfitness	Not Related tothe Major andMinor

IKS	IndianKnowledgeSystem	I. Generic IKS Course:basicknowledgeoftheIKS II. Subject Specific IKSCourses: advancedinformation pertaining to the subject:partofthemajor credit.	Subject SpecificIKS related toMajor
OJT	On-Job Training(Internship/Apprenticeship)	correspondingtotheMajorSubject	Relatedtothe Major
FP	Fieldprojects	correspondingtotheMajor Subject	Relatedtothe Major
CC	Co-curricularCourses	Health and Wellness, Yogaeducation sports, andfitness, Cultural Activities,NSS/NCC and Fine/Applied/Visual/Performing Arts	Not Related tothe Major andMinor
CE	Community Engagementandservice		NotRelatedtotheMajorand Minor
RP	ResearchProject	correspondingtotheMajor Subject	Relatedtothe Major

Programme Template:

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

(Major) discipline-related knowledge acquisition, mastery over cognitive and professional, vocational skills are to be used e.g. demonstrate sound understanding of, analyses, compare, create, design, etc…(minimum5)		Mathematics, Operating system and Networking.
	2.	Understand the ethical and professional responsibilities in the field of computer applications by adhering to professional standards and practices.
	3.	Applying programming knowledge to develop a software application to solve specific problems.
	4.	Analyzing system requirements to design efficient and effective software solutions.
	5.	Evaluate software designs and architectures for efficiency, security and user experience.
	6.	Create a software application to meet the requirements of the Industrial Standards.
Eligibility Criteria for Programme		A candidate for being eligible for admission to the course leading to the degree of Bachelor of Computer Applications (as per revised NEP 2020) must have passed the Higher Secondary School Certificate (Std. XII) examination conducted by the different Divisional Boards of the Maharashtra State Board of Secondary and Higher Secondary Education with 45% marks (40% for candidates belonging to Reserved category) with the following subjects: - (from any stream). • English • Any one of the Modern Indian Languages or Modern Foreign Languages or any classical Language or Information Technology/ Any four subjects carrying 100 marks each. Candidate from Maharashtra - MAH-B.BCA/BBA/BMS/BBM -CET entrance test is mandatory From outside Maharashtra – CUET (Central University Entrance Test) OR • English • Any one vocational subject carrying 200 marks • Any three subjects carrying 100 marks each. OR Must have passed the Higher Secondary School Certificate (Std. XII) examination with the Minimum Competency based vocational courses (MCVC) conducted by the different Divisional Boards of the Maharashtra State Board of Secondary and Higher Secondary Education. • English • Any one of the Modern Indian Languages or Modern

		Foreign Languages or any classical. • Language or Information Technology • General Foundation Course. • Any one subject carrying 300 marks from among the Minimum Competency based vocational courses prescribed by the Higher Secondary School Certificate examination from time to time. OR Must have passed an examination of other Board or Body Recognized as equivalent thereto. OR 1. Must have passed any three year Government recognized Diploma programme. <u>B.C.A.II Year :</u> • 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 12 credits of major(core) courses in the first year and passed first year without ATKT. • Should have passed three years Govt./ D.T.E. recognized Diploma Course with First Class and without ATKT. B.C.A.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 28 credits of major(core) courses in the first and second year and passed first and second year without ATKT. <u>B.C.A.IV Year:</u> • For a student from our university /another university- should have completed three years degree in the same discipline.
Intake (For SNDTWU Departments and Conducted Colleges)		-

- *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	L	T	P	Credits	Marks	Int	Ext
	Semester I								
1.1	Problem Solving Techniques	Major(Core)	2	0	4	4	100	50	50
1.2	Mathematics Foundations to Computer Science - I	Major(Core)	2	0	0	2	50	0	50
1.3	Digital Marketing	OEC	2	0	4	4	100	50	50
1.4	Computer Architecture	VSC	1	0	2	2	50	0	50
1.5	Web Technology - I	SEC	1	0	2	2	50	50	0
1.6	General English - I	AEC	2	0	0	2	50	0	50
1.7	Indian Knowledge System	IKS	2	0	0	2	50	50	0
1.8	Professional Ethics	VEC	2	0	0	2	50	0	50
1.9	* Co-Curricular Course	CC	2	0	0	2	50	50	0
						22	**550**	**250**	**300**
	*** Co-Curricular Course (Health & Wellness, Yoga education, sports & fitness, Cultural activities, NSS, NCC and Fine/applied/visual/performing arts)**								
	Semester II		**L**	**T**	**P**	**Credits**	**Marks**	**Int**	**Ext**
2.1	Data Structures	Major(Core)	2	0	4	4	100	50	50
2.2	Mathematics Foundations to Computer Science - II	Major(Core)	2	0	0	2	50	0	50
2.3	Programming with Python or Fundamentals of Python Programming	Minor Stream	1	0	2	2	50	0	50
2.4	Intellectual Property Rights	OEC	4	0	0	4	100	50	50
2.5	Operating Systems	VSC	1	0	2	2	50	0	50
2.6	Object Oriented Programming using Java	SEC	1	0	2	2	50	50	0
2.7	General English - II	AEC	2	0	0	2	50	0	50
2.8	Environmental Studies and Sustainability	VEC	2	0	0	2	50	0	50
2.9	* Co-Curricular Course	CC	2	0	0	2	50	50	0
						22	**550**	**250**	**300**
	*** Co-Curricular Course (Health & Wellness, Yoga education, sports & fitness, Cultural activities, NSS, NCC and Fine/applied/visual/performing arts)**								

ExitwithUGCertificatewith10extracredits(44+10credits)

Course Syllabus Semester I

1.1 Major (Core)

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

	Characteristics of Algorithms, Standard Algorithm Format. **Problems Involving Iteration and Nesting:** Displaying Different Patterns and Shapes Using Symbols and Numbers, Generating Arithmetic and Geometric Progression, Fibonacci and Other Sequences, Approximate Values For π, Sin(x), Cos(x), Etc. Using Taylor Series. Different Kinds of Data in The Real World and How They are Represented in The Computer Memory. **Representation of Integers:** Signed Magnitude Form, 1's Complement And 2's Complement. Representation of Real Numbers: IEEE 754 Floating Point Representation. Representation of Characters: ASCII, UNICODE. **C Language:** Introduction To Programming Languages, Different Generations of Programming Languages. Typed Vs Typeless Programming Languages, History of C Language, An Empty C Program. C Language Counterparts For Input (scanf()), Output (printf()) Statements, Assignment, Arithmetic, Relational and Logical Operators. If, If-Else Statements, For, While, Do-While Statements. Data Types. Translating Pseudocode/Algorithm to C Program. Incremental Compilation and Testing of The C Program. Simple Problems Involving Input, Output, Assignment Statement, Selection and Repetition. Good Coding Practices.
Module 3 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	Apply logical reasoning and mathematical operations to solve a variety of number-based problems, such as extracting digits, identifying palindromes, calculating prime factors, detecting special numbers and converting numbers between bases
	Use advanced control structures in C, including the else-if ladder and switch cases, and employ increment/decrement operators, along with break and continue statements.
	Perform statistical operations
Content Outline	**Problems on Numbers:** Extracting Digits of a Number (Left to Right and Right to Left), Palindrome, Prime Number, Prime Factors, Amicable Number, Perfect Number, Armstrong Number, Factorial, Converting Number from One Base to Another. Statistics (Maximum, Minimum, Sum and Average) on a Sequence of Numbers which are Read using Sentinel- Controlled Repetition using only a few Variables. **C Language**: else-if Ladder, switch Case, Increment/Decrement Operators, break and continue Statements, structure, pointers
Module 4 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	develop modular programs using both top-down and bottom-up approaches
	apply recursive functions to solve complex problems, gaining an understanding of when recursion is appropriate.
Content Outline	Modular Programming, Top-Down and Bottom-Up Approaches to Problem Solving. Recursion. Problems on Arrays: Reading and Writing of Array Elements, Maximum, Minimum, Sum, Average, Median and Mode. Sequential And Binary Search. Any one Sorting Algorithm. Matrix Operations. **C Language:** Function Definition and Declaration (Prototype), Role of Return Statement, One Dimensional and Two-Dimensional Arrays. String Functions. Other Operators, Operator Precedence and Associativity. Debugging.

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

1. Assignments can be given range from simple tasks to complex projects.
2. Evaluate their problem-solving approach, algorithm design, and logic.
3. Conducting regular quizzes and tests to assess students' theoretical knowledge of C programming concepts

Course Syllabus

References:-

Text Books

1. Venkatesh, Nagaraju Y, Practical C Programming for Problem Solving, Khanna BookPublishing Company, 2024.
2. AICTE's Programming for Problem Solving (with Lab Manual), Khanna Book PublishingCompany, 2024.
3. Harvey Deitel and Paul Deitel, C How to Program, 9th edition, Pearson India, 2015.
4. R G Dromey, How to Solve It by Computer.

Reference Books

1. Brian W. Kernighan and Dennis Ritchie, The C Programming Language, 2nd edition, Pearson, 2015.
2. Jeri Hanly and Elliot Koffman, Problem Solving and Program Design in C, 8th edition, Pearson,2015.

Problem Solving Techniques: Lab Problems

UNIT-II

1. Converting degrees Celsius to Fahrenheit and vice versa?
2. Display three input numbers in sorted (non-decreasing) order?
3. Given a positive integer value n (>= 0) display number, square and cube ofnumbers from 1 to n in a tabular format?
4. Given an input positive integer number, display odd numbers from in therange[1,n]?
5. Display first mathematical tables, each table up to 10 rows? Generalise this todisplayfirst n (> 0) mathematical tables up to m (m > 0) rows?
6. Display following patterns of n rows (n > 0), For the below examples n = 5?Foreach pattern write a separate algorithm/program?

$ $$ $$$ $$$$ $$$$$	$ $$ $$$ $$$$ $$$$$	12345 1234 123 12 1	12345 1234 123 12 1

7. Display the following patterns of n rows (n > 0), for the below examples n = 5?

Hollow square pattern:	Triangle Patterns with numbers:	Square with diagonals:	Diamond Pattern
#####	1	* * * * *	*
# #	121	* * * *	***
# #	12321	* * *	*****
# #	1234321	* * * *	***
#####	123454321	* * * * *	*

8. Given the first term (a), difference/multiplier (d) and number of terms (n > 0),display the first n terms of the arithmetic/geometric progression?
9. Display the first n (n > 0) terms of the fibonacci sequence?
10. Display the first n (n > 0) terms of the Tribonacci sequence?
11. Given two positive integer numbers n1 and n2 check if the numbers areconsecutive numbers of the fibonacci sequence?
12. Compute approximate value of π considering first n (n > 0) terms of theTaylor **series for π?**
13. Compute approximate value of e^x considering first n (n > 0) terms of theTaylor series for e^x?
14. Compute approximate value of sin(x)/cos(x) considering first n (n > 0) terms oftheTaylor series for sin(x)/cos(x)?

UNIT-III

1. Extract digits of an integer number (left to right and right to left)?
2. Given a sequence of digits form the number composed of the digits. Use sentinel controlled repetition to read the digits followed by -1. For example, forthe input 2 7 32 9 -1 the output number is 27329?
3. Check if a given positive integer number is a palindrome or not?
4. Compute character grade from the marks (0 ≤ marks ≤ 100) of a subject. Grading Scheme: 80-100 : A, 60 - 79: B, 50 - 59: C, 40-49: D, 0-39: F? Solve this using both else-if ladder and switch case?
5. Compute the sum of a sequence of numbers entered using sentinel controlled

repetition?

6. Check if a given positive integer number is a prime number or not?
7. Compute prime factors of a positive integer number?
8. Check if two positive integer numbers are amicable numbers or not?
9. Check if a given positive integer number is a perfect number or not?
10. Check if a given positive integer number Armstrong number or not?
11. Converting a positive integer number (n > 0) from one base (inputBase) to another base (outputBase) (2 <= input Base, outputBase <= 10). Input number should be validated before converting to make sure the number uses only digits allowed in the input base?
12. Write a program to display a number in text form. For example If the number is 5432the output should be "FIVE FOUR THREE TWO"?
13. Using the grading scheme described in the question 4 (UNIT III), Compute how manystudents awarded each grade and display the frequency as a bar chart (horizontal) using single "*" for each student. Use sentinel controlled repetition (-1 as sentinel value) inreading the students marks. Use else-if ladder/switch case to compute the grade and the corresponding frequency.

    Sample bar chart when the class has 7-A, 10-B, 3-C, 7-D and 1-F grades.

    A:
    ****
    B:
    ****
    C: ***
    D:
    ****
    F: *
14. Compute maximum, minimum, sum and average of a sequence of numbers which areread using sentinel controlled repetition using only few variables?
15. Compute body mass index, BMI = weight in KGs / (Height in Meters ***Height in Meters), Both weight and height values are positive real numbers. Your**

    program should display BMI value followed by whether the person is Underweight,Normal, Overweight or Obese using the below ranges:
    BMI Values
    Underweight: less
    than 18.5Normal:
    >=18.5 and <25
    Overweight: >=25 and < 30
    Obese: >= 30

UNIT IV

1. Design a modularized algorithm/program to check if a given positive integer numberis a circular prime or not?
2. Design a modularized algorithm/program to compute a maximum of 8 numbers?
3. Design a modular algorithm/program which reads an array of n integer elements andoutputs mean (average), range (max-min) and mode (most frequent elements)?
4. Design a modular algorithm/program which reads an array of n integer elements andoutputs median?
5. Implement your own string length and string reversal functions?
6. Design algorithm/program to perform matrix operations addition, subtractionandtranspose?
7. Write a recursive program to count the number of digits of a positive integernumber?
8. Recursive solutions for the following problems:
    a. Factorial of a number?
    b. Display digits of a number from left to right (and right to left)?
    c. Compute x^y using only multiplication?
    d. To print a sequence of numbers entered using sentinel controlledrepetition inreverse order?

Course Syllabus Semester I

1.2 Major (Core)

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

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

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

- Problem-Solving Assignments
- Conceptual Quizzes

Course Syllabus

Text Books

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

Reference Books

1. Grimaldi Ralph P. and Ramana B. V., Discrete and Combinatorial Mathematics: AnApplied Introduction, Fifth Edition, Pearson Education, 2007.
2. Rosen Kenneth H. and Krithivasan Kamala, Discrete Mathematics and its Applications,McGraw Hill, India, 2019.
3. West Douglas B., Introduction to Graph Theory, Second Edition, Pearson Education,2015

Web Resources

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

1.3 OEC

Course Title	**Digital Marketing**
Course Credits	**4 Credits**
Course Outcomes	**After Completion of this Course, students will be able**
	CO1: Understand the fundamental concepts of digital marketing, including its role in the modern marketing mix.
	CO2: Develop skills in creating and managing digital marketing strategies using various tools and platforms.
	CO3: Implement effective online campaigns through SEO, social media, email marketing, and other digital channels.
	CO4: Analyze key performance metrics and optimize marketing strategies for better results
	CO5: Understand emerging trends in digital marketing, such as AI, mobile marketing, and content marketing.
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	Understanding the Fundamentals and Scope of Digital Marketing
	Developing and Implementing a Digital Marketing Strategy
	Setting and Measuring Digital Marketing Goals and Performance
Content Outline	**Introduction to Digital Marketing:** Definition and scope of digital marketing, Importance and growth of digital marketing in modern business, Comparison between traditional and digital marketing, The digital marketing ecosystem (Search engines, social media, email marketing, etc.), Key terminologies (CTR, CPC, CPA, ROI, etc.) **Digital Marketing Strategy and Planning:** Developing a digital marketing strategy, understanding buyer personas and target audience, building a customer journey and funnel, Importance of content in digital marketing (content marketing), Integration of digital channels (SEO, social media, Email, PPC), Setting SMART goals and KPIs for digital marketing
Module 2 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	Understand SEO Principles and Website Optimization
	Develop Effective Social Media Marketing Strategies
	Measure and Analyse Marketing Performance:

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

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

- Case Study Analysis
- PPC Campaign Simulation
- Performance Metrics Report
- Group Presentation on Affiliate Marketing Strategies
- Google Analytics Hands-On Exercise
- Weekly Quizzes
- Final Project Presentation

Course syllabus

References: -

Text Books & Reference Books

1. Bhatia, Puneet Singh. *Fundamentals of Digital Marketing*. Pearson Education India, 2017.
2. Chaffey, Dave, and Fiona Ellis-Chadwick. *Digital Marketing: Strategy, Implementation, and Practice*. Indian ed., Pearson Education India, 2020.
3. Gupta, Seema. *Digital Marketing: Text and Cases*. McGraw Hill Education India, 2017.
4. Ahuja, Vandana. *Digital Marketing: A Practical Approach*. 2nd ed., Oxford University Press India, 2015.
5. Dodson, Ian. *The Art of Digital Marketing: The Definitive Guide to Creating Strategic, Targeted, and Measurable Online Campaigns*. Wiley India, 2016.
6. Kaur, Bhavneet. *Introduction to Digital Marketing*. BPB Publications, 2020.
7. Das, Abhishek. *Digital Marketing: Concepts, Strategy and Practice*. Himalaya Publishing House, 2020.
8. Mehta, Rohit. *SEO & Social Media Marketing: A Comprehensive Guide to Optimize Websites, Increase Traffic, and Engage Users*. Rohit Mehta Publishing, 2019.

Unit 1

Short Answer Questions:

1. Define digital marketing and explain how it differs from traditional marketing.
2. List and explain the key components of digital marketing.
3. What are the advantages of digital marketing over traditional marketing?
4. Define the following terms: CTR, CPC, CPA, ROI.
5. Explain the role of search engines in digital marketing.
6. How does social media contribute to digital marketing strategies?
7. What is a digital marketing strategy, and why is it important?
8. Define and explain the concept of a "buyer persona."
9. What are SMART goals, and how are they used in digital marketing?
10. What is a customer journey, and why is it critical in digital marketing planning?
11. Explain the importance of KPIs (Key Performance Indicators) in digital marketing.

Unit II

Long Answer Questions:

1. Discuss the importance of digital marketing in today's business environment. How has it transformed marketing practices globally?
2. Explain the digital marketing ecosystem and describe how the different components (SEO, social media, Email Marketing) work together.
3. Discuss the process of developing a digital marketing strategy from the ground up. What are the key elements to consider?
4. How does content marketing contribute to the success of a digital marketing plan? Provide examples of effective content strategies.
5. Analyse the role of integration across different digital channels (SEO, social media, Email) in a successful campaign.

Unit II/III

1. Explain the role of keywords in SEO. How do they influence search engine rankings?
2. What are the most important factors affecting a website's SEO performance?
3. Discuss the significance of mobile optimization in SEO. Why is it essential in today's digital landscape?
4. Compare and contrast organic social media marketing and paid social media advertising. What are the pros and cons of each?
5. Explain the importance of engagement on social media platforms. How does it affect brand visibility and customer relationships?
6. Discuss the role of influencers in social media marketing. How can they contribute to a brand's strategy?

Unit IV

1. Discuss the advantages of email marketing over other forms of digital marketing.
2. What are some common reasons for low open rates in email campaigns, and how can they be improved?
3. Explain the importance of personalization in email marketing. How does it impact customer engagement and conversion rates?

Unit IV

Practical/Hands-on Questions:

1. Conduct a basic SEO audit on a website using Google Search Console. What areas would you focus on, and why?
2. Use a keyword research tool (like SEMrush or Google Keyword Planner) to find five keywords relevant to a chosen website. How would you optimize a webpage for these keywords?
3. Examine a webpage and list five on-page SEO improvements you can suggest to increase its visibility on search engines.
4. What are the key differences between on-page and off-page SEO? Give examples of how each can be implemented.
5. Set up a Google Ads search campaign for a hypothetical product or service. Outline the key steps involved, from keyword selection to ad creation.
6. Use Google Analytics to track the performance of a website. What key metrics (like bounce rate, session duration) would you analyse?
7. Analyse the performance of a PPC campaign. What are the essential metrics (CPC, CTR, conversion rate) that you would monitor to evaluate success?
8. Create a performance report for a marketing campaign. Include metrics such as traffic sources, conversion rates, and ROI.
9. Create a social media content calendar for a brand for the next 7 days. Include post ideas and platforms.
10. Design a mock Facebook or Instagram ad using an online ad tool. What audience targeting options would you use?
11. Use a social media scheduling tool (like Buffer or Hootsuite) to schedule posts for one week. What factors should you consider when scheduling?
12. How would you evaluate the performance of a social media campaign? List the key metrics you would analyse.
13. Create a sample email marketing campaign using Mailchimp or a similar platform. Include subject lines, body text, and a call-to-action (CTA).
14. Segment an email list based on user behaviour or demographics. Explain how segmentation can improve email marketing performance.

15. Develop an automated email series for a new product launch. What types of emails would you include, and when would you send them?
16. Analyse the performance of an email marketing campaign. What metrics (like open rates, click-through rates) would you focus on, and how would you optimize the campaign?
17. Set up a Google Ads search campaign for a hypothetical product or service. Outline the key steps involved, from keyword selection to ad creation.
18. Use Google Analytics to track the performance of a website. What key metrics (like bounce rate, session duration) would you analyse?
19. Analyse the performance of a PPC campaign. What are the essential metrics (CPC, CTR, conversion rate) that you would monitor to evaluate success?
20. Create a performance report for a marketing campaign. Include metrics such as traffic sources, conversion rates, and ROI.

Case Study Questions:

1. Analyse a case study of a successful digital marketing campaign (real or hypothetical). Identify the key components that contributed to its success and how the strategy could be improved further.
2. Create a digital marketing strategy for a small business launching a new product. Include SEO, social media, email marketing, and PPC in your plan.
3. Imagine a company is experiencing a decline in website traffic. What steps would you take to diagnose and solve the issue?
4. A company's email marketing campaign is not delivering the desired results in terms of conversions. Suggest changes to improve the campaign's effectiveness.

Text Books and Reference Books

1. Bhatia, Puneet Singh. *Fundamentals of Digital Marketing*. Pearson Education India, 2017.
2. Chaffey, Dave, and Fiona Ellis-Chadwick. *Digital Marketing: Strategy, Implementation, and Practice*. Indian ed., Pearson Education India, 2020.
3. Gupta, Seema. *Digital Marketing: Text and Cases*. McGraw Hill Education India, 2017.
4. Ahuja, Vandana. *Digital Marketing: A Practical Approach*. 2nd ed., Oxford University Press India, 2015.
5. Dodson, Ian. *The Art of Digital Marketing: The Definitive Guide to Creating Strategic, Targeted, and Measurable Online Campaigns*. Wiley India, 2016.
6. Kaur, Bhavneet. *Introduction to Digital Marketing*. BPB Publications, 2020.
7. Das, Abhishek. *Digital Marketing: Concepts, Strategy and Practice*. Himalaya Publishing House, 2020.
8. Mehta, Rohit. *SEO & Social Media Marketing: A Comprehensive Guide to Optimize Websites, Increase Traffic, and Engage Users*. Rohit Mehta Publishing, 2019.

1.4 VSC

Course Title	**Computer Architecture**
Course Credits	**2 Credits**
Course Outcomes	**After Completion of this Course, students will be able**
	CO1: To Understand the basics of Digital Electronics and Binary Number System
	CO2: To Learn the implementation of Combinational Circuit
	CO3: To Learn the implementation of Sequential Circuit
	CO4: To Understand the Organization of basic computers.
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	Understand Digital Signals and Computer Architecture
	Apply Boolean Algebra and Karnaugh Maps for Simplification
	Convert and Performing Arithmetic Operations in Various Number Systems
Content Outline	**Digital Principles**: Introduction to Digital Signals, Definition and characteristics of digital signals vs. analog signals, Basic understanding of digital logic and digital computers. **Von Neumann Architecture**: Overview of Von Neumann architecture and its significance in digital systems. **Boolean Laws and Theorems**: Introduction to basic Boolean algebra, K-Map (Karnaugh Map): Focus on 2 and 3-variable K-Maps for simplification, Simplification of Boolean expressions using SOP (Sum of Products) and POS (Product of Sums) forms, Don't Care Conditions for optimization of K-Maps (at a basic level). **Basic Number Systems:** Overview of Decimal, Binary, Octal, and Hexadecimal number systems, Basic Number System Conversions between binary, decimal, octal, and hexadecimal. **Binary Arithmetic:** Simple binary arithmetic (addition and subtraction). Binary and Decimal Codes: **ASCII and Gray Code basics, Brief introduction to Error-detecting codes.**
Module 2 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	Understand Combinational Circuit Design
	Explore Sequential Circuits and Flip-Flop
	Utilize Shift Registers and Binary Counter
Content Outline	**Combinational and Sequential Circuits:** Half Adder and Full Adder: Basic design and working, Multiplexer and Demultiplexer: Overview and practical applications. **Sequential Circuits:** Flip-Flops: SR Flip-Flop and D Flip-Flop. **Shift Registers:** Introduction to 4-bit Shift Registers and their practical uses. Binary Counters: I**ntroduction to a 4-bit synchronous binary counter.**

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

- Group Presentation on Applications of Combinational and Sequential Circuits
- Weekly Quizzes

Course Syllabus

Text Books

1. Donald P Leach, Albert Paul Malvino, Goutam Saha- "Digital Principles &Applications" , Tata McGraw Hill Education Private Limited,2011Edition.
2. M. Morris Mano- "Computer System Architecture", Pearson/Phi, Third Edition.

Reference Books

1. William Stallings- "Computer Organization and Architecture", Pearson/PHI, SixthEdition,
2. Andrew S. Tanenbaum- "Structured Computer Organization", PHI /Pearson 4th Edition,
3. M.V .Subramanyam, "Switching Theory and Logic Design", Laxmi Publications (P)Ltd.
4. Ikvinderpal Singh, Computer Organization Architecture, Khanna Book Publishing.

Computer Architecture

Unit I: Digital Logic and Basic Gates:

1. Verify the logic behaviour of AND, OR, NAND, NOR, EX-OR, EX-NOR, Invert, and Buffer gates.
2. Study and verify NAND as a Universal Gate.
3. Verify De Morgan's theorem for 2 variables.
4. Design and implement the 4:1 MUX using gates.
5. Realize Decoder and Encoder circuit using Basic Gates.
6. Convert Binary to Grey Code.

Unit II: Combinational and Sequential Circuits:

7. Design and test of an S-R flip-flop using NAND/NOR gates.
8. Verification of Truth Tables of J-K Flip-Flop using NAND/NOR gate.
9. Design and verify the operation of a half adder and full adder.
10. Design and verify the operation of a half subtractor.
11. Design and implement a 4-bit synchronous counter.
12. Design and verify the 4-bit asynchronous counter.
13. Design and implement a 4-bit shift register using Flip-flops.
14. Convert BCD to Excess-3 code using NAND gates.
15. Implementation of 4-Bit Parallel Adder Using 7483 IC.
16. Implement Boolean function using logic gates in both SOP and POS.

Hardware Experiments

1. Familiarize the computer system layout:
2. Identify the Computer Name and Hardware Specifications:
3. Identify and Troubleshoot problems with RAM, SMPS, and the motherboard.
4. Configure BIOS settings:
5. Add additional RAM to the system:

6. Study the motherboard layout of a system:
7. Demonstrate the assembly of a PC:
8. Demonstrate various ports on the CPU:
9. Install and configure Windows OS:
10. Study the installation of a printer and troubleshoot common printer problems.

1.5 SEC

Course Title	**Web Technology - I**
Course Credits	**2 Credits**
Course Outcomes	**After Completion of this Course, students will be able**
	Understand the fundamental concepts of web technologies and the Internet.
	Gain proficiency in creating static and dynamic web pages using HTML, CSS
	To develop skills in creating forms and multimedia content.
	To learn layout techniques for creating responsive and visually appealing web designs.
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	Understand and 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**
	Understand the Basics and Importance 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.

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

- Web Page Design Project
- CSS Styling Exercise
- Presentation on Web Standards and Accessibility

Course Syllabus

Text Books

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

Reference Books

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

Web Technology

1. What does HTML stand for? Explain its purpose.
2. Describe the structure of a basic HTML document.
3. What are the key differences between block-level and inline elements? Give examples of each.
4. How do you create a hyperlink in HTML? Provide an example.
5. What are semantic HTML elements? Give three examples and their purposes.
6. Explain the purpose of the <head> section in an HTML document. What kind of elements can you include there?
7. How can you include an image in an HTML document? Describe the <img> tag and its attributes.
8. What is the difference between and elements?
9. Describe how to create an HTML form. What are some common input types?
10. Explain the significance of the alt attribute in the <img> tag.
11. What is the difference between the <table>, <thead>, <tbody>, and <tfoot> elements in HTML?
12. How can you embed multimedia content in an HTML document? Describe the <audio> and <video> tags and their attributes.
13. Explain how to create a responsive web page using HTML5 features. What elements would you use?
14. What are data attributes in HTML? Provide an example of how they can be used.
15. Discuss the importance of using the <meta> tag in HTML. What types of metadata can it provide?
16. What does CSS stand for, and what is its primary purpose?
17. Explain the difference between inline, internal, and external CSS. Provide examples.
18. What is a CSS selector? Describe the difference between class selectors and ID selectors.
19. How can you apply styles to an HTML element using CSS? Provide an example.
20. Explain the use of pseudo-classes and pseudo-elements in CSS. Provide examples of each.
21. What are CSS properties for setting text styles? Name at least five.
22. Describe the different ways to specify colour in CSS. Provide examples of each method.
23. Explain the process of using CSS transitions and animations. Provide examples of when they might be useful.

Practical Questions

1) Create Personal Portfolio Page

Task: Create a personal portfolio webpage that includes:

- A header with your name and a brief tagline.
- An about section describing yourself.
- A projects section that lists at least three projects, each with a title, description, and a link to view the project.
- A contact section with a form to collect user input (name, email, message).

2) Create Image Gallery

Task: Create a simple image gallery that displays:

- A grid of images.
- Each image should be a link to an enlarged version when clicked (you can use a placeholder link).
- Include captions for each image.

3) Create Landing Page

Task: Design a landing page for a fictional product that includes:

- A captivating headline and a brief description.
- Call-to-action buttons (e.g., "Buy Now" or "Learn More").
- Testimonials from users.
- An email signup form for a newsletter.

4) User Feedback Form

Task: Develop a user feedback form that includes:

- Fields for name, email, and feedback comments.
- A rating system using radio buttons (1 to 5 stars).
- A thank-you message displayed after submission (you can use static text).

5) FAQ Section

Task: Develop a Frequently Asked Questions (FAQ) section that includes:

- A list of questions with expandable answers using <details> and <summary> tags for better user experience.

6) Create Simple Table

Task: Create a simple HTML table to display your daily schedule.
Include:

a. Columns for Time, Activity, and Duration.
b. At least three rows of data.

7) Create Styling Tables

Task: Create a table to display data (e.g., a comparison of smartphone features) and apply CSS styles to:

- Change the background colour of the header.
- Alternate row colours.
- Add padding to cells for better readability.

8) Create Order Form

Task: Create an order form for a fictional product that includes:

- Fields for Name, Address, Product Selection (dropdown), Quantity, and Credit Card Information.
- A Submit button.
- Implement basic validation for required fields.

9) Create File Upload Form

Task: Create a form that allows users to upload a file. Include:

a. An input field for Name.
b. A file input for uploading documents or images.
c. A Submit button.
d. Implement basic validation to limit file types and sizes.

10) Create Basic Hyperlink Creation

Task: Create a webpage with a simple hyperlink that leads to your favorite website.
Ensure the link opens in a new tab.

11) Create Anchor Links

Task: Build a single-page website with anchor links.
Include sections like Home, About, and Contact.
Use anchor links to navigate between these sections smoothly.

1.6 AEC

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

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

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

- Writing Assignments
  Enhance descriptive and explanatory writing skills.
- Oral Communication Activities (Language Lab or Interactive Settings)
  Improve active listening and comprehension skills.

Course Syllabus

References:-

Text Books & Reference Books

1. AICTE's Prescribed Textbook: Communication Skills in English (with Lab Manual),Anjana Tiwari, Khanna Book Publishing Co., 2023.
2. Effective Communication Skills. Kul Bhushan Kumar, Khanna Book Publishing, 2022.
3. Practical English Usage. Michael Swan. OUP. 1995.
4. Remedial English Grammar. F.T. Wood. Macmillan.2007
5. On Writing Well. William Zinsser. Harper Resource Book. 2001
6. Study Writing. Liz Hamp-Lyons and Ben Heasly. Cambridge University Press. 2006.
7. Communication Skills. Sanjay Kumar and PushpLata. Oxford University Press. 2011. 8.Exercises in Spoken English. Parts. I-III. CIEFL, Hyderabad. Oxford University Press.

1.7 IKS

https://sndt.ac.in/pdf/academics/syllabus-as-per-nep/iks-syllabus/ug-degree/inception-of-indian-knowledge-system.pdf

1.8 VEC

https://sndt.ac.in/nep2020/syllabus-as-per-nep/vec-syllabus

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

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

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

1) Provide students with real-life ethical case studies from various professional fields.
2) Assign students to research and present on specific ethical dilemmas that professionals commonly encounter in their fields.
3) Organize group discussions or debates on contemporary ethical issues related to different professional fields.
4) Assign students to work in groups and develop an ethical leadership project aimed at promoting ethical conduct in a specific context, such as a workplace, community organization, or educational institution.

Course Syllabus

References:-

Text Books

1. **"Ethics in Information Technology"** by George Reynolds
2. **"Computer Ethics: A Global Perspective"** by Richard Spinello
3. **"Ethics for the Information Age"** by Michael J. Quinn

Reference Books

1. **"Computers and Society"** by Charles B. Perkins
2. **"The Ethics of Information Technology and Business"** by Richard A. Spinello
3. **"The Responsible Administrator: An Approach to the Ethics of the Profession"** by Terry L. Cooper

Course Syllabus Semester: I

1.9 CC

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

Course Syllabus

Semester: II

2.1 Major (Core)

Course Title	**Data Structures**
Course Credits	**4 credits**
Course Outcomes	CO1: Understand the fundamental concepts of Data Structures and their applications
	CO2: Develop problem-solving skills using Data Structures.
	CO3: Implement Data Structures using C programming language.
Module 1 (Credit 1)	
Learning Outcomes	**Learners will be able to**
	Define Data Structures and describe their role in organizing, managing, and storing data efficiently for various applications.
	Explain Basic Data Structure Operations, including creation, insertion, deletion, updating, and traversal.
	Define Multi-Dimensional Arrays and explain their representation in memory for applications that require higher-dimensional data organization.
	Apply Linear and Binary Search Techniques to search for elements within an array and compare their efficiency.
Content Outline	**Introduction and Overview:** Definition, Classification and Operations of Data Structures. Algorithms: Complexity, Time-Space Tradeoff. **Arrays:** Definition and Classification of Arrays, Representation of Linear Arrays in Memory, Operations on Linear Arrays: Traversing, Inserting, Deleting, Searching, Sorting and Merging. Searching: Linear Search and Binary Search, Comparison of Methods. Sorting: Bubble Sort,Selection Sort, and Insertion Sort. Two-Dimensional Arrays, Representation of Two- Dimensional Arrays in Memory, Matrices and Sparse Matrices, Multi-Dimensional Arrays.
Module 2 (Credit 1)	
Learning Outcomes	**Learners will be able to**
	Define Linked Lists and explain their structure, highlighting the differences between linked lists and arrays in terms of memory allocation, flexibility, and performance.
	Understand Applications of Linked Lists
	Define Hashing and its importance in data retrieval, explaining the concept of hash functions.
	Explain the Concept of Collision in hashing, discussing the implications of multiple keys hashing to the same index in a hash table.

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

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

- Written Assignments
- Practical Programming Assignments
- Projects
- Group Activities
- Lab Activities
- Quizzes and Tests

References/Textbooks:-

Text Books

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

Reference Books

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

Web Resources

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

Lab Programs:

1. Write a program for insertion and deletion operations in an array.
2. Write a program to search for an element in an array using Linear Search and BinarySearch.
3. Write a program to sort an array using Bubble Sort, Selection Sort and Insertion Sort.
4. Write a program to merge two arrays.
5. Write a program to add and subtract two matrices.
6. Write a program to multiply two matrices.
7. Write a program to insert an element into a Singly Linked List:
   (a) At the beginning
   (b) At the end
   (c) At a specified position
8. Write a program to delete an element from a Singly Linked List:
   (a) At the beginning
   (b) At the end
   (c) A specified element

9. Write a program to perform the following operations in a Doubly Linked List:
    (a) Create
    (b) Search for an element
10. Write a program to perform the following operations in a Circular Linked List:
    (a) Create
    (b) Delete an element from the end
11. Write a program to implement stack operations using an array.
12. Write a program to implement stack operations using a linked list.
13. Write a program to add two polynomials using a linked lists.
14. Write a program to evaluate a postfix expression using a stack.
15. Write a program to perform the following using recursion:
    (a) Find the factorial of a number
    (b) Find the GCD of two numbers
    (c) Solve Towers of Hanoi problem
16. Write a program to implement simple queue operations using an array.
17. Write a program to implement circular queue operations using an array.
18. Write a program to implement circular queue operations using a linked list.
19. Write a program to perform the following operations on a binary search tree.
    (a) Preorder Traversal
    (b) Inorder Traversal
    (c) Postorder Traversal
20. Write a program to perform insertion operation in a binary search tree.

Course Syllabus

Semester: II

2.2 Major (Core)

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

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

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

- Solve a few problems where you can compute these errors based on provided data and analyze how impact the accuracy of results in numerical methods.
- Solve a set of algebraic and transcendental equations using both the Bisection Method and the Newton-Raphson Method.
- Formulate a real-life optimization problem as a linear programming problem. Solve it using both the Graphical Method (for two-variable problems) and the Simplex Method (for larger problems).

References/Text Books:-

Text Books

1. Kolman B., Busby R. and Ross S., Discrete Mathematical Structures, 6th Edition,Pearson Education, 2015.
2. Sastry S. S., Introductory Methods of Numerical Analysis, Fifth Edition, PHL, 2022.
3. Taha Hamdy A., Operations Research: An Introduction, Eighth Edition, PearsonPrentice Hall, 2003.
4. S.B. Singh, Discrete Structures, Khanna Book Publishing, 2023 (AICTERecommended Textbook)

Reference Books

1. Rosen Kenneth H. and Krithivasan Kamala, Discrete Mathematics and itsApplications, McGraw Hill, India, 2019.

2. Chakravorty J. G. and Ghosh P. R., Linear Programming and Game Theory, MoulikLibrary, 2017.
3. Sharma J. K., Operations Research: Theory and Applications, Fourth Edition,Macmilllan Publishers, 2007.

Web Resources

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

Course Syllabus

Semester: II

2.3 Minor Stream

Course Title	**Programming with Python**
Course Credits	**2 Credits**
Course Outcomes	**CO1**: Gain proficiency in the basics of Python programming
	CO2: Develop problem-solving skills using programming.
	CO3: Understand and apply fundamental programming concepts such as variables, loops, and functions.
	CO4: Write, debug, and test Python programs effectively.
Module 1 (Credit 1)	
Learning Outcomes	**Learners will be able to:**
	Apply Applications of Python and Set Up a Python Development Environment
	Understand and Write Basic Python Code and work with Variables and Data Types
	Use Conditional Statements for Decision-Making
	Implement Loops for Repetition
Content Outline	**Introduction to Python and Basic Syntax:** • Overview of Python and its applications • Setting up Python environment (IDEs, Jupyter Notebooks) • Understanding Python syntax and structure • Variables and data types (integers, floats, strings) **Control Flow – Conditionals and Loops** • Boolean expressions • Conditional statements (if, elif, else) • Introduction to loops (for, while) • Controlling loops with break and continue
Module 2 (Credit 1)	
Learning Outcomes	**Learners will be able to:**
	Use and Manipulate Lists and Understand and Apply Tuples
	Implement and Manage Dictionaries and define and Call Functions
	Understand Variable Scope and Implement Basic Error Handling

Content Outline	**Data Structures – Lists, Tuples, and Dictionaries** • Lists: indexing, slicing, and list methods • Tuples and differences from lists • Dictionaries and key-value pairs **Functions and Modular Programming** • Defining and calling functions • Function arguments and return values • Scope of variables • Basic error handling (try, except)

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

- Weekly Assignments and Practical Exercises
- Quizzes and Short Tests
- Group Coding Activities and Peer Review
- Problem-Solving Exercises and Mini-Projects
- Final Project

Textbooks:

- *Programming and Problem Solving with Python* by Ashok Namdev Kamthane and Amit Ashok Kamthane
- *Core Python Programming* by R. Nageswara Rao
- *Python Programming: A Modular Approach* by Sheetal Taneja and Naveen Kumar
- *Python Programming* by Reema Thareja
- *Let Us Python* by Yashavant Kanetkar
- *Python Made Easy* by Sarika Jain

References:-

- Kamthane, A. N., & Kamthane, A. A. (2018). *Programming and Problem Solving with Python*. Pearson Education India.
- Rao, R. N. (2018). *Core Python Programming*. Dreamtech Press.

Short Questions (1-2 Marks)

1. What are the key features of Python that make it popular?
2. Define variables and data types in Python.
3. What is the difference between = and == in Python?
4. How can you take user input in Python?
5. Explain the purpose of the print() function.
6. What does IDE stand for, and name any two IDEs used for Python development.
7. How do you create a single-line comment in Python?
8. What is the difference between a list and a tuple?
9. How can you check the data type of a variable in Python?
10. What is the purpose of the len() function?
11. What are the basic data types available in Python?
12. How do you convert a string to an integer in Python?
13. What is the output of 5 // 2 and explain why.
14. Define what a dictionary is in Python.
15. Explain the purpose of the break statement in a loop.
16. What is None in Python, and how is it used?
17. How does the continue statement work in a loop?
18. Define the term "function" in programming.
19. Explain the difference between local and global variables.
20. What is meant by "exception handling," and why is it used?
21. How do you declare a function in Python?
22. What is a lambda function?
23. Write a simple lambda function to add two numbers.
24. Explain how to open a file in read mode in Python.
25. What does the try and except block do in Python?
26. What is self in Python classes?
27. How do you create an infinite loop in Python?
28. Describe what the range() function does.
29. What is the output of list(range(1, 10, 2))?
30. How can you reverse a list in Python?

Long Questions (5-10 Marks)

1. Explain Python's data types. Discuss numbers, strings, lists, tuples, dictionaries, and sets with examples.
2. Describe control flow statements in Python, including if, elif, and else statements. Provide code examples demonstrating their usage.
3. Explain loops in Python. Discuss for and while loops with examples, and explain the role of break and continue statements within loops.
4. Discuss Python's list operations. Explain indexing, slicing, appending, and other list methods with examples.
5. Explain the difference between mutable and immutable types in Python with examples. Discuss why strings are immutable and lists are mutable.
6. Write a Python program to find the factorial of a number using both recursion and iteration. Explain the differences in approach.
7. Describe functions in Python. How do you define and call a function? Discuss function parameters, return values, and give examples.
8. Explain the concept of scope in Python. Illustrate with examples the difference between local and global variables and the global keyword.
9. What is a dictionary in Python? Describe dictionary operations, such as adding, updating, deleting elements, and accessing keys/values with examples.
10. Explain exception handling in Python. What are the try, except, else, and finally statements, and how do they work together? Provide examples.

11. Discuss list comprehensions in Python. What is their syntax, and how do they differ from traditional for-loops? Provide examples with conditions.
12. Explain file handling in Python. Write code to open, read, and write to files and explain the modes (r, w, a, r+) used in file operations.
13. Discuss object-oriented programming (OOP) in Python. Define class, object, inheritance, and polymorphism with examples.
14. Explain constructors and destructors in Python classes. Write code that uses a constructor to initialize class attributes and a destructor to clean up.
15. Write a program to sort a list of dictionaries by a specific key in each dictionary (e.g., sort students by grade). Explain the logic.
16. Describe lambda functions in Python and explain how they differ from regular functions. Provide examples using lambda functions with map, filter, and reduce.
17. Explain modules in Python. What is the purpose of a module, and how do you import modules? Explain the use of import, from ... import, and as.
18. Write a Python program to create a simple calculator. The calculator should be able to handle basic operations like addition, subtraction, multiplication, and division.
19. Explain error types in Python (SyntaxError, TypeError, ValueError, etc.) with examples for each.
20. Create a class in Python to represent a bank account. Include methods to deposit, withdraw, and check balance. Ensure withdrawals cannot exceed the available balance and explain the program logic.

Project-Based Questions (10-15 Marks)

- Write a Python program that performs basic arithmetic operations (addition, subtraction, multiplication, and division) based on user input.
- Create a program that checks if a number entered by the user is even or odd.
- Write a program to calculate the area of different shapes (circle, square, rectangle) based on user input.
- Write a Python program that converts temperatures between Celsius and Fahrenheit.
- Create a Python program to check if a given number is prime or not.
- Write a program that calculates simple interest based on principal, rate, and time entered by the user.
- Write a program to find the factorial of a number using both recursion and iteration.
- Create a program that checks if a given year is a leap year
- Write a program that checks if a given string is a palindrome.
- Write a Python program that reverses a given list without using built-in functions
- Write a program to find the sum of all elements in a list.
- Write a program to find the sum of all elements in a list.
- Write a Python program to merge two dictionaries into one.
- Write a program to perform matrix addition and multiplication based on user input.
- Write a function that finds and returns all prime numbers in a given range.
- Build a simple calculator program using functions for each arithmetic operation.

Course SyllabusSemester: II

2.4 OEC

Course Title	**Intellectual Property Rights**
Course Credits	**4 Credits**
Course Outcomes	**CO1**: Identify different types of Intellectual Property and understand their significance.
	CO2: Explain the legal and procedural aspects of acquiring and protecting IPR.
	CO3: Analyze case studies to understand infringement, enforcement, and dispute resolution in IPR.
	CO4: Demonstrate the ability to draft documents related to IP registration and management.
	CO5: Apply IP strategies in business, research, and technology transfer scenarios.
Module 1 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	Define and clearly explain the purpose and types of IPR, understanding their importance to individuals, organizations, and society.
	Recognize the historical development and ethical justifications for IPR and discuss these within economic and social contexts.
	Identify and describe the major international treaties and organizations that govern IP protection worldwide.
	Analyse how regional agreements and national legislation shape the protection, enforcement, and limitations of IPR globally. Apply knowledge of IPR types, treaties, and laws in real-world examples and case studies.
Content Outline	**Unit 1:** **Introduction to Intellectual Property Rights** • Overview of IPR: Meaning, purpose, and importance. • Types of Intellectual Property: Patents, Copyrights, Trademarks, Trade Secrets, Geographical Indications, Industrial Designs. • History and Evolution of IPR. • Economic and Ethical Justifications for IPR. **Legal Framework and International Conventions** • Overview of International IP Treaties: Paris Convention, Berne Convention, TRIPS, WIPO. • IPR Legislation: National and International IP laws. • Role of WIPO and WTO in IPR. Regional Agreements and their Impact on IP.
Module 2 (Credit 1)	
Learning Outcomes	After learning the module, learners will be able to
	Understanding Patent and Copyright Fundamentals

	Applying Procedures for Filing and Enforcement
	Conducting Patent Searches and Recognizing Fair Use
Content Outline	**Unit 2: Patents** • Introduction to Patents: Definition, scope, and types. • Patentable and Non-Patentable Inventions. • Patent Filing Procedure: National and International filing (PCT). • Patent Infringement and Enforcement: Remedies and Case Studies. • Patent Search and Database: Hands-on exercise in patent search tools. **Copyrights and Related Rights** • Introduction to Copyright: Definition, scope, and duration. • Authorship and Ownership. • Registration and Transfer of Copyrights. • Copyright Infringement and Fair Use. • Case Studies in Copyright Infringement.
Module 3 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	Understanding Trademarks and Industrial Designs
	Recognizing the Importance and Management of Trade Secrets and Geographical Indications
	Analyzing Emerging Issues in Intellectual Property
Content Outline	**Unit 3:** **Trademarks and Industrial Designs** • Introduction to Trademarks: Definition, scope, and types. • Trademark Registration Process: National and International registration (Madrid System). • Trademark Infringement, Passing Off, and Remedies. • Industrial Designs: Concept, protection, registration, and case studies. **Trade Secrets, Geographical Indications, and Emerging Issues** • Trade Secrets: Definition, protection, and management. • Geographical Indications: Concept, registration, and case studies. • Emerging Issues in IPR: Biotechnology, Software Patents, Traditional Knowledge, Domain Names.
Module 4 (Credit 1)	
Learning Outcomes	**After learning the module, learners will be able to**
	Understanding IPR Commercialization and Strategic Management
	Recognizing Ethical Issues and Societal Impacts of IPR
	Analyzing Global IPR Challenges and Dispute Resolutions
Content Outline	**Unit 4:** **IPR Management and Strategy** • Commercialization of IPR: Licensing, Assignment • IPR in Research and Development: Innovation and Technology Management.

	• IPR for Startups and Entrepreneurs. **IPR, Ethics, and Global Challenges** • Ethical Issues in IPR: Plagiarism, Patent Trolls • IPR and Innovation in a Global Economy. • Impact of IPR on Society, Technology, and Business. • Global Disputes and Resolutions: Case Studies.

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

- Case Study Analysis
- IP Search and Patent Drafting
- Group Presentation
- Role-Playing Activity
- Workshop: Copyright & Digital Media
- Ethical Issues Essay
- Trademark Creation Exercise
- Project Work: IPR in Startups
- Quiz and Short Tests

Recommended Textbooks:

1. Intellectual Property Rights: Unleashing the Knowledge Economy by Prabuddha Ganguli.
2. Intellectual Property Law by Lionel Bently and Brad Sherman.
3. WIPO Intellectual Property Handbook by World Intellectual Property Organization.
4. Patent Law and Policy by Robert Patrick Merges and John Fitzgerald Duffy.
5. Cyber law by Vivek Sood

References:

1. Intellectual Property Rights in India by G. B. Reddy.
2. Managing Intellectual Property by David Bainbridge.
3. Intellectual Property Rights: Law and Practice by Harish Chander.

Web Resources:

- WIPO (World Intellectual Property Organization): www.wipo.int
- USPTO (United States Patent and Trademark Office): www.uspto.gov
- EPO (European Patent Office): www.epo.org

Question Bank: Intellectual Property Rights (IPR)

General IPR Questions

Basic Level

1. What is Intellectual Property (IP)?
2. List and briefly describe the different types of Intellectual Property.
3. What is the role of WIPO (World Intellectual Property Organization)?
4. Define "infringement" in the context of IPR.
5. What is the significance of TRIPS (Trade-Related Aspects of Intellectual Property Rights)?
6. Differentiate between "Intellectual Property" and "Physical Property."
7. What are the economic benefits of protecting Intellectual Property?
8. Define "Patent Cooperation Treaty (PCT)."
9. What is meant by "Public Domain"?
10. How does IPR encourage innovation?

Intermediate Level

1. Explain the concept of "IPR Lifecycle."
2. What is the purpose of "IP Licensing"?
3. What are the international treaties related to Copyright protection?

4. Explain "Doctrine of Exhaustion" in IPR.
5. What is "Compulsory Licensing" and in what situations can it be applied?
6. Discuss the ethical considerations related to IPR.
7. How can companies protect their trade secrets?
8. What are the implications of violating IPR?
9. Discuss the role of IPR in the digital economy.

Advanced Level

1. Analyze the impact of globalization on IPR.
2. What is "Patent Troll" and how does it affect innovation?
3. Compare and contrast "Fair Use" and "Infringement."
4. How does IPR affect developing countries differently compared to developed countries?
5. Discuss the balance between access to knowledge and IPR protection.
6. What is the "Doctrine of Equivalents" in patent law?
7. Explain how Artificial Intelligence (AI) and Machine Learning are challenging existing IPR frameworks.
8. Analyze the role of IPR in biotechnology and pharmaceuticals.
9. Discuss the implications of open-source software on traditional IPR.
10. What are the challenges in protecting Traditional Knowledge through IPR?

Patent Questions

Basic Level

1. What is a Patent?
2. What are the criteria for an invention to be patentable?
3. Define "Prior Art."
4. What is the difference between a Utility Patent and a Design Patent?
5. What is a "Patent Claim"?
6. How long does patent protection last?
7. What is meant by "Patent Pending"?
8. What is a "Provisional Patent Application"?
9. Describe the "Patent Filing Process."

Intermediate Level

1. What are the grounds for patent rejection?
2. Discuss the concept of "Patent Infringement."
3. What is the significance of the "Patent Search Report"?
4. How does the "Patent Cooperation Treaty (PCT)" simplify international patent filing?
5. What is "Patent Opposition" and when can it be filed?
6. Explain the concept of "Patent Valuation."
7. Discuss the procedure for transferring patent ownership.
8. What are "Patent Pools" and how do they function?
9. Describe the differences between "Product Patent" and "Process Patent."
10. Explain the role of "Compulsory Licensing" in patents.

Advanced Level

1. Analyze a case study of a famous patent infringement lawsuit.
2. Explain "Evergreening of Patents" and its implications in the pharmaceutical industry.
3. What is "Patent Thicket" and how does it impact the market?
4. Discuss the challenges of patenting software and algorithms.
5. Evaluate the effectiveness of the TRIPS agreement in harmonizing patent laws.
6. Discuss the impact of the "America Invents Act" on the US patent system.

Copyright Questions

Basic Level

1. What is Copyright?
2. What are the main rights provided by copyright law?
3. How long does copyright protection last?
4. What is "Public Domain"?
5. Define "Fair Use" in copyright law.
6. What is the difference between a "Copyright" and a "Patent"?
7. What is a "Derivative Work"?
8. How is copyright infringement handled?
9. What is meant by "Moral Rights" in copyright?
10. Explain the term "Digital Rights Management (DRM)."

Intermediate Level

1. What are the exceptions to copyright protection?
2. How can one register a copyright?
3. What is the significance of the Berne Convention in copyright law?
4. Discuss "Copyright in the Digital Age."
5. What is "Plagiarism" and how does it differ from copyright infringement?
6. Explain the concept of "Orphan Works" in copyright law.
7. What are "Neighbouring Rights" in the context of copyright?
8. How does copyright apply to software and digital content?
9. What is the "Three-Step Test" in copyright?
10. Discuss the legal implications of sharing copyrighted content online.

Advanced Level

1. Analyze a famous copyright infringement case and its implications.
2. Discuss the concept of "Copyright Exhaustion" and its global impact.
3. How is copyright protection managed for traditional and folk art?
4. What are the challenges of enforcing copyright in a global digital environment?
5. Explain the impact of copyright laws on the entertainment industry.
6. Discuss the legal challenges of remix culture and user-generated content.
7. What are the ethical considerations in copyrighting educational materials?
8. How do copyright laws impact access to scientific publications?
9. Discuss the concept of "Creative Commons" and its importance.
10. Analyze the effectiveness of copyright enforcement mechanisms.

Trademark Questions

Basic Level

1. What is a Trademark?
2. What are the benefits of registering a trademark?
3. What is "Trademark Infringement"?
4. What is the difference between a "Trademark" and a "Service Mark"?
5. How long does trademark protection last?
6. Define "Passing Off" in the context of trademark law.
7. What is a "Collective Trademark"?
8. How can a trademark be renewed?
9. What is meant by "Distinctiveness" in trademarks?
10. Explain the process of "Trademark Registration."

Intermediate Level

1. What is the Madrid System for international trademark registration?
2. Discuss the criteria for a valid trademark.
3. What is "Trademark Dilution"?
4. Explain the concept of "Domain Name Disputes."
5. What are the remedies available for trademark infringement?

6. How does a trademark differ from a trade name?
7. What is "Certification Mark" in trademark law?
8. Discuss the role of "Geographical Indications" as a form of trademark.
9. What is "Trademark Assignment" and how is it executed?
10. Explain the legal implications of using a similar mark in different industries.

Advanced Level

1. Analyze a famous trademark dispute case.
2. What is the significance of "Non-Traditional Trademarks"?
3. How can brand identity be protected through trademark law?
4. Discuss the global challenges in trademark protection for multinational companies.
5. How does "Parallel Import" impact trademark owners?
6. What are the legal and economic implications of "Trademark Counterfeiting"?
7. Evaluate the role of trademarks in building brand value.
8. How can businesses effectively manage a global trademark portfolio?
9. Discuss the impact of social media on trademark enforcement.

Trade Secret Questions

Basic Level

1. What is a Trade Secret?
2. What are the key elements that qualify information as a trade secret?
3. How does a trade secret differ from a patent?
4. What are some examples of trade secrets?
5. What is "Misappropriation" in trade secret law?
6. How can a company protect its trade secrets?
7. Define "Confidentiality Agreement."
8. What is the role of Non-Disclosure Agreements (NDAs) in protecting trade secrets?
9. Explain the legal consequences of trade secret theft.
10. What are the advantages and disadvantages of protecting information as a trade secret?

Intermediate Level

1. Discuss the importance of trade secrets in a competitive business environment.
2. What is the role of employees in maintaining trade secret confidentiality?
3. What is "Doctrine of Inevitable Disclosure" in trade secret law?
4. Explain the concept of "Economic Espionage."
5. Discuss the implications of trade secret theft in the global economy.
6. How does the Uniform Trade Secrets Act (UTSA) standardize trade secret protection?
7. What are the remedies available for trade secret misappropriation?
8. What are the differences between a Non-Compete Agreement and a Non-Disclosure Agreement?
9. Discuss the challenges of protecting trade secrets in a digital environment.

Advanced Level

1. Analyze a major trade secret theft case.
2. Discuss the balance between employee mobility and trade secret protection.

Basic Level

1. What is the difference between IP Licensing and IP Assignment?
2. Define "IP Commercialization" in the context of Intellectual Property Rights.
3. What are the primary types of licensing agreements?
4. Explain the term "Exclusive License" in the context of IPR.
5. What is a "Non-exclusive License"?
6. How can businesses benefit from IP Assignment?
7. What are "Royalties" in the context of IP Licensing?
8. What is meant by "Territoriality" in IP Licensing?

Intermediate Level

1. Discuss the advantages and disadvantages of exclusive licensing for a patent holder.
2. What is the significance of "Field of Use" restrictions in licensing agreements?
3. How does "Sublicensing" work in the context of IP?
4. What are the common challenges faced during the commercialization of IPR?

IPR in Research and Development: Innovation and Technology Management

Basic Level

1. How does IPR support research and development (R&D)?
2. Define "Innovation Management" in the context of IPR.
3. What role does IPR play in technology transfer?
4. What is a "Technology Transfer Office"?
5. Explain the concept of "R&D Agreements" related to IPR.
6. What is "Innovation Disclosure" and how is it related to IPR?
7. How can IPR contribute to the protection of new technologies?
8. What are the benefits of protecting IP assets during the R&D phase?

Intermediate Level

1. Discuss the importance of IP strategy in technology management.
2. What are the challenges of managing IP in collaborative research projects?
3. Explain the concept of "Open Innovation" and how it relates to IPR.
4. What is the role of patents in protecting new technology developed through R&D?
5. Discuss the concept of "IP Risk Management" in R&D.

Advanced Level

1. Evaluate the role of IPR in fostering innovation within a competitive market.
2. Discuss the impact of "Trade Secrets" on research and innovation.
3. How does the management of IP in R&D differ across various industries (e.g., pharmaceuticals vs. IT)?
4. Discuss the challenges of protecting traditional knowledge in modern R&D settings.

IPR for Startups and Entrepreneurs

Basic Level

1. Why is IPR important for startups and entrepreneurs?
2. What are the key IPR challenges faced by startups?
3. How can a startup protect its brand using trademarks?
4. What is the importance of filing a provisional patent for startups?

Intermediate Level

1. What are the common pitfalls in managing IPR for startups?
2. How can startups use IP Licensing as a revenue source?
3. Discuss the role of IPR in securing venture capital funding.

Advanced Level

1. Analyse a case study where strong IPR protection led to a startup's success.
2. What are the ethical dilemmas faced by startups in aggressively defending their IP?

Question Bank for Unit 4: IPR, Ethics, and Global Challenges

Ethical Issues in IPR: Plagiarism, Patent Trolls

Basic Level

1. What is Plagiarism in the context of IPR?
2. Define "Patent Troll."
3. What are the common ethical issues associated with IPR?
4. How does plagiarism differ from copyright infringement?
5. What is the impact of Patent Trolls on innovation?
6. Why is ethical consideration important in IPR?

7. What is the role of ethics in IPR enforcement?
8. How does plagiarism affect academic and research integrity?
9. What are some examples of unethical behavior in IPR?
10. What steps can be taken to prevent plagiarism in academic research?

Intermediate Level

1. Discuss the legal and ethical implications of patent trolling.
2. What are the ethical concerns related to IP hoarding?
3. Discuss the challenges of detecting plagiarism in research.

Advanced Level

1. Analyse the impact of patent trolling on small businesses and startups.
2. Discuss a case study involving significant plagiarism in academia.

IPR and Innovation in a Global Economy

Basic Level

1. How does IPR contribute to global innovation?
2. What is the role of WIPO in the global IPR landscape?
3. How do international IPR agreements influence innovation?
4. What challenges do multinational companies face with IPR protection?

Course Syllabus
Semester: II

2.5 VSC

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

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

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

- Weekly Quizzes and Tests
- Coding Challenges
- Practical Lab Sessions
- OS Simulation and Mini Projects
- Seminar and Presentation

Text Books:

1. Ekta Walia, Operating Systems Concepts, Khanna Publishing House, 2022 (AICTERecommended Textbook)
2. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne (2006), Operating System Principles, 7th edition OR Later edition, Wiley India Private Limited, New Delhi.
3. Stallings (2006), Operating Systems, Internals and Design Principles, 5th edition, Pearson Education, India.

Reference Books:

1. Andrew S Tanenbaum, Modern Operating Systems, Third Edition, Prentice Hall India.

Question Bank
Operating Systems
Basic Level

1. Define an Operating System (OS).
2. What are the main components of an Operating System?
3. Describe the services provided by an Operating System.
4. What is a Batch Operating System?
5. Explain the concept of Multiprogramming.
6. What is Time Sharing in Operating Systems?
7. Differentiate between Parallel, Distributed, and Real-Time Systems.
8. What is the kernel in the context of an OS?
9. List the different types of Operating Systems.
10. What are the advantages of a Time-Sharing System?

Intermediate Level

1. Explain the structure and architecture of an Operating System.
2. How do the roles of a System Call and System Program differ in an OS?
3. What are the benefits and limitations of Distributed Systems?
4. How does a Real-Time Operating System differ from a general-purpose OS?
5. Explain the concept of a System Call with an example.
6. What are the advantages of Multiprogramming over Batch Systems?

Advanced Level

1. Discuss the evolution of Operating Systems from Batch Systems to Real-Time Systems.
2. How does the architecture of a Distributed OS differ from a centralized OS?
3. analyse the impact of OS services on system performance.
4. Discuss how an OS manages hardware resources in a Multiprogramming environment.

Process Management

Basic Level

1. What is a Process?
2. List and explain the different states of a Process.
3. Define a Process Control Block (PCB).
4. What is the difference between a Program and a Process?
5. Explain the concept of Threads.
6. What are the advantages of using Threads?
7. Differentiate between User-Level Threads and Kernel-Level Threads.
8. What is Context Switching?
9. Describe what happens during a Process State Transition.
10. What is the purpose of Process Scheduling?

Intermediate Level

1. Explain the difference between a single-threaded and multi-threaded process.
2. What is a Process Scheduling Algorithm? Why is it important?
3. Describe the structure of a Process Control Block (PCB).
4. How do processes communicate with each other in an OS?
5. Explain the term "Multithreading" and its benefits.
6. What is the role of a Scheduler in an OS?
7. Discuss the difference between Cooperative and Preemptive Scheduling.
8. Explain the concept of Inter-Process Communication (IPC).
9. How does an OS handle multiple threads within the same process?
10. What are the common challenges in Process Management?

Advanced Level

1. Compare and contrast Process-based and Thread-based multitasking.
2. Discuss the advantages of Kernel-Level Threads over User-Level Threads.
3. How does an OS handle deadlocks in Process Scheduling?

Process Scheduling

Basic Level

1. What is Process Scheduling?
2. What are the main objectives of Process Scheduling?
3. Define Preemptive Scheduling.
4. Define Non-Preemptive Scheduling.
5. Explain the First-Come, First-Served (FCFS) Scheduling Algorithm.
6. What is the Shortest Job First (SJF) Scheduling Algorithm?
7. How does Round Robin (RR) Scheduling work?
8. What is a CPU Burst in Process Scheduling?
9. What are Scheduling Criteria?
10. List the types of Process Schedulers.

Intermediate Level

1. Compare Preemptive and Non-Preemptive Scheduling.
2. How does Priority Scheduling work, and what are its limitations?
3. Explain the concept of Time Quantum in Round Robin Scheduling.
4. Discuss the advantages and disadvantages of the FCFS Scheduling Algorithm.
5. How can the Aging technique solve the problem of Starvation in Scheduling?
6. What is Turnaround Time, and how is it calculated?
7. Compare the performance of SJF and Round Robin Scheduling Algorithms.
8. What is Throughput in the context of Scheduling?
9. Discuss how Scheduling Algorithms impact CPU utilization.

Advanced Level

1. Analyze the trade-offs between different Scheduling Algorithms for real-time systems.
2. Discuss how Load Balancing is managed in Multi-Processor Scheduling.
3. Evaluate the impact of Time Quantum size in Round Robin Scheduling on system performance.

Unit II: Process Synchronization, Deadlocks, Memory Management, Virtual Memory, I/O Management

Process Synchronization

Basic Level

1. What is Process Synchronization?
2. Explain Inter-Process Communication (IPC).
3. What is a Race Condition?
4. Define the Critical Section Problem.
5. What is Mutual Exclusion?
6. What are Semaphores?
7. Explain the concept of a Monitor in synchronization.
8. What is a Busy Waiting problem?
9. How can Deadlocks occur during process synchronization?
10. What is the role of Synchronization in Multithreading?

Intermediate Level

1. Discuss the significance of the Critical Section in Process Synchronization.
2. Discuss the role of Locks in Process Synchronization.
3. Explain how Priority Inversion can be handled.

Advanced Level

1. Analyze the differences between Mutexes and Semaphores.
2. Discuss how the OS manages Synchronization in Real-Time Systems.
3. Compare different IPC mechanisms (e.g., Pipes, Message Queues, Shared Memory).
4. What are the challenges of achieving Process Synchronization in multi-core systems?
5. How can Deadlock be detected and prevented in Process Synchronization?

Deadlocks

Basic Level

1. What is a Deadlock?
2. Explain the four necessary conditions for a Deadlock.
3. What is the System Model in Deadlocks?
4. What is a Safe State in Deadlock prevention?
5. How can Deadlocks be avoided?
6. Define Deadlock Detection.
7. What is the Banker's Algorithm?
8. Explain Resource Allocation Graphs.
9. What are Deadlock Prevention techniques?
10. How can a Deadlock be recovered?

Intermediate Level

1. What is the difference between Deadlock Prevention and Deadlock Avoidance?
2. Explain how the Banker's Algorithm works.
3. How can Deadlocks be detected in a multi-resource system?
4. Explain how Deadlock Avoidance can affect system performance.

Advanced Level

1. Analyze a case study involving Deadlocks in a Distributed System.
2. Evaluate the impact of Deadlocks on system performance.
3. How does Deadlock Detection work in a distributed environment?
4. What are the limitations of the Banker's Algorithm?
5. How does the OS handle Deadlocks in modern multi-threaded environments?
6. Compare different Deadlock Handling techniques in Operating Systems.

Operating Systems LAB

List of experiments

1. Write C program to simulate the FCFS CPU Scheduling algorithm.
2. Write C program to simulate the SJF CPU Scheduling algorithm.
3. Write C program to simulate the Round Robin CPU Scheduling algorithm.
4. Write a C program to simulate Bankers Algorithm for Deadlock Avoidance.
5. Write a C program to implement the Producer – Consumer problem using semaphores.
6. Write a C program to illustrate the IPC mechanism using Pipes.
7. Write a C program to illustrate the IPC mechanism using FIFOs.
8. Write a C program to simulate Paging memory management technique.
9. Write a C program to simulate Segmentation memory management technique.
10. Write a C program to simulate the Best Fit contiguous memory allocation technique.
11. Write a C program to simulate the First Fit contiguous memory allocation technique.
12. Write a C program to simulate the concept of Dining-Philosophers problem.
13. Write a C program to simulate the MVT algorithm.
14. Write a C program to implement FIFO page replacement technique.
15. Write a C program to write a C program for implementing sequential file allocation method.

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

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

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

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

Text Books

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

Reference Books

1. Arunesh Goyal, The Essentials of JAVA, Khanna Book Publishing Company PrivateLimited, 2012.
2. Tanweer Alam, Core JAVA, Khanna Book Publishing Company Private Limited, 2015.
3. Y. Daniel Liang, Introduction to Java Programming, 7th Edition, Pearson,2008.
4. S. Malhotra and S. Choudhary, Programming in Java, 2nd Edition, OxfordUniversityPress, 2014.

Web Resources

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

Short Questions

1. What is Object-Oriented Programming (OOP)? List its main features.
2. What is the difference between a class and an object in Java?
3. Explain encapsulation with an example.
4. What is inheritance, and why is it useful in OOP?
5. How does polymorphism work in Java? Provide a brief example.
6. What is an interface in Java? How does it differ from an abstract class?
7. Define method overloading and method overriding. How are they different?
8. What is a constructor in Java? What are its types?
9. Explain the final keyword in Java. What are its uses?
10. What is exception handling in Java? Why is it important?
11. What is the purpose of the this keyword in Java?
12. How do you implement multiple inheritance in Java?

13. What is the difference between ArrayList and LinkedList in Java?
14. What is a Java package, and how do you create one?
15. How does garbage collection work in Java?
16. What is the significance of the static keyword in Java?
17. What is an abstract class in Java? Provide an example.
18. How does the super keyword function in Java?
19. Explain the difference between == and .equals() in Java.
20. What is a HashMap in Java, and when should it be used?

Long Questions:

- Explain the four pillars of OOP (Encapsulation, Inheritance, Polymorphism, Abstraction) with examples in Java.
- Discuss the difference between interfaces and abstract classes in Java. Provide a scenario where each would be appropriate.
- How does Java handle exceptions? Write a program demonstrating exception handling using try, catch, and finally.
- Describe the Java Collection Framework. Explain the different types of collections (List, Set, Map) with examples.
- What is multithreading in Java? Discuss its advantages and provide a sample program using threads.
- Explain how inheritance works in Java with an example. What are the types of inheritance supported?
- Discuss how Java handles file input and output. Write a program that reads from a file and writes to another file.
- How do you implement polymorphism in Java? Discuss both method overloading and method overriding with examples.
- Describe the concept of design patterns in OOP. Explain any two design patterns with examples in Java.
- How do you achieve data abstraction in Java? Provide an example using abstract classes and interfaces.
- What are Java Generics, and why are they important? Give examples of generic classes and methods.
- Explain how the synchronized keyword works in Java. Write a program demonstrating thread synchronization.
- What is JDBC, and how is it used in Java? Explain with an example of connecting to a database.
- How does Java's garbage collector work? Discuss the different garbage collection strategies used in Java.
- Discuss the differences between String, StringBuilder, and String Buffer in Java.
- What is the difference between checked and unchecked exceptions in Java? Provide examples.
- Explain how Java handles concurrency. Discuss the Executor framework with an example.
- How do you manage access control in Java? Explain the different access modifiers with examples.
- What is reflection in Java? Discuss its applications and provide a simple example.
- Explain the concept of serialization and deserialization in Java. Write a program to serialize and deserialize an object.

List of Practical:

1. Write a program to read two numbers from user and print their product.
2. Write a program to print the square of a number passed through command line arguments.
3. Write a program to send the name and surname of a student through command line arguments andprint a welcome message for the student.
4. Write a java program to find the largest number out of n natural numbers.
5. Write a java program to find the Fibonacci series & Factorial of a number using recursive and nonrecursive functions.
6. Write a java program to multiply two given matrices.
7. Write a Java program for sorting a given list of names in ascending order.
8. Write a Java program that checks whether a given string is a palindrome or not . Ex:MADAM is apalindrome.
9. Write a java program to read n number of values in an array and display it in reverse order.
10. Write a Java program to perform mathematical operations. Create a class called AddSub withmethods to add and subtract. Create another class called MulDiv that extends from AddSub class to use the member data of the superclass. MulDiv should have methods to multiply and divide A main function should access the methods and perform the mathematical operations.
11. Create a JAVA class called Student with the following details as variables within it.
    - USN, NAME, BRANCH, PHONE, PERCENTAGE
    - Write a JAVA program to create n Student objects and print the USN, Name, Branch, Phone,and percentage of these objects with suitable headings.
12. Write a Java program that displays the number of characters, lines and wordsin a text.
13. Write a Java program to create a class called Shape with methods called getPerimeter() andgetArea(). Create a subclass called Circle that overrides the getPerimeter() and getArea()

methods to calculate the area and perimeterof a circle.

14. Write a Java program to create a class Employee with a method called calculateSalary(). Createtwo subclasses Manager and Programmer. In eachsubclass, override the calculateSalary() method to calculate and return the salary based on their specific roles.
15. Write a Java program using an interface called ‘Bank’ having function ‘rate_of_interest()’. Implement this interface to create two separate bank classes ‘SBI’ and ‘PNB’ to print different rates of interest. Include additionalmember variables, constructors also in classes ‘SBI’ and ‘PNB’.
16. Write a Java package program for the class book and then import the datafrom the package anddisplay the result.
17. Write a Java program for finding the cube of a number using a package for various data typesand then import it in another class and display the results.
18. Write a Java program for demonstrating the divide by zero exceptionhandling.
19. Write a Java program that reads a list of integers from the user and throws anexception if any numbers are duplicates.
20. Create an exception subclass UnderAge, which prints “Under Age” along with the age value when an object of UnderAge class is printed in the catch statement. Write a class exceptionDemo in which the method test() throws UnderAge exception if the variable age passed to it as argument is less than 18. Write main() method also to show working of the program.

Course Syllabus

Semester: II

2.7 AEC

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

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

Assignments/Activities towards Comprehensive Continuous Evaluation (CCE)

- Assignment on Job Application and CV
- Business emails (To be sent by students to the teacher as mentioned in syllabus)
- Report writing on any given topic.

References: -

1) Urmila Rai, S M Rai, "Business Communications", Himalaya Publishing House, 2004
2) Business Correspondence and Report Writing by Prof. R.C. Sharma & Krishna Mohan, Tata McGraw Hill & Co. Ltd., 2001, New Delhi
3) Business Correspondence and Report Writing by Prof. R.C. Sharma & Krishna Mohan, Tata McGraw Hill & Co. Ltd., 2001, New Delhi

Course Syllabus
Semester: II

1.8 VEC (Environmental Awareness)

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

Course Syllabus Semester: II

2.9 Co-Curricular Course

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

Evaluation and Grades

* Lab Components are practical oriented and no theory examinations will be conducted. Practical examinations will be conducted and evaluated by Internal and External Examiners.
• Int. Exam: C.A.: Internal examination and continuous assessment involves two internal test + Assignments/ Presentation/ Oral/ Viva/Group Discussion etc

Terms Used:

- Tu.: Tutorial
- Pr.: Practical
- C.A.: Continuous Assessment
- Tw.: Term Work
- Cr.: Credits
- Int.: Internal
- Ext.: External

NEP Structure and Credit Score
The undergraduate degree under NEP will be a multi-disciplinary program of four-year duration with multiple exit and entry options.

Description	Certificate Type	Minimum Credits Required
The successful completion of 2 semesters	Certificate	44
The successful completion of 4 semesters	Diploma	88
The successful completion of 6 semesters	Degree	132
The successful completion of 8 semesters	Honours	176