SWAMI RAMANAND TEERTH MARATHWADA UNIVERSITY, NANDED - 431 606 (MS)

(Credit Framework and Structure of

Bachelor of Computer Application (Single Major)

(BCA)

First Year

with Multiple Entry and Exit Options as per NEP-2020)

UNDERGRADUATE PROGRAMME OF

SCIENCE & TECHNOLOGY

Major in **BCA** and Minor in **DSM** (Computer Application)

Swami Ramanand Teerth Marathwada University, Nanded

Faculty of Science and Technology (Three Optional in the First Year)

Credit Framework for Four Year Multidisciplinary Degree Program with Multiple Entry and Exit

Subject: BCA (Major) /DSM (Minor 1 and Minor 2)

Bachelor of Computer Application (Single Major) First Year

Eligibility:12th Arts/ Commerce/Science/ MCVC

Year & Level	Semester	Optional 1 (Major) *(From the same Faculty)*	Optional 2 (Minor 1) *(From the same Faculty)*	Optional 3 (Minor 2) *(From the same Faculty)*	Generic Elective (GE) (select from Basket 3 of *Faculties other than Science and Technology*)	Vocational & Skill Enhancement Course	Ability Enhancement Course (AEC) (Basket 4) Value Education Courses (VEC) / Indian Knowledge System (IKS) (Basket 5) *(Common across all faculties)*	Field Work / Project/Internship/ OJT/ Apprenticeship / Case Study Or Co-curricular Courses (CCC) (Basket 6 for CCC) *(Common across all faculties)*	Credits	Total Credits
1	2	3	4	5	6	7	8	9	10	11
1 (4.5)	I	**SBCACT1101** (T 2Cr) **SBCACP1101** (P 2Cr) **4 Credits**	**SBCACT1101** (T 2Cr) **SBCACP1101** (P 2Cr) **4 Credits**	**SBCACT1101** (T 2Cr) **SBCACP1101** (P 2Cr) **4 Credits**	**SBCAGE1101 2 Credits**	**SBCASC1101 2 Credits**	**AECENG1101** (2Cr) **ACEMIL1101** (2Cr) **IKSXXX1101** (2Cr) **6 Credits**		**22**	**44**
	II	**SBCACT1151** (T 2Cr) **SBCACP1151** (P 2Cr) **4 Credits**	**SBCACT1151** (T 2Cr) **SBCACP1151** (P 2Cr) **4 Credits**	**SBCACT1151** (T 2Cr) **SBCACP1151** (P 2Cr) **4 Credits**	**SBCAGE1151 2 Credits**	**SBCASC1151 2 Credits**	**AECENG1151** (2Cr) **ACEMIL1151** (2Cr) **VECCOI1151** (2Cr) ***Constitution of India*** **6 Credits**		**22**	
	Cum. Cr.	**08**	**08**	**08**	**04**	**04**	**08**	**04**	**44**	

***Exit option*: UG Certificate in Opt 1, Opt 2 and Opt 3 on completion of 44 credits** and additional **4 credits from NSQF / Internship**

Abbreviations:

1. **DSC:** Department/Discipline Specific Core (Major)
2. **DSE:** Department/Discipline Specific Elective (Major)
3. **DSM:** Discipline Specific Minor
4. **GE/OE:** Generic/Open Elective
5. **VSEC:** Vocational Skill and Skill Enhancement Course
6. **VSC:** Vocational Skill Courses
7. **SEC:** Skill Enhancement Courses
8. **AEC:** Ability Enhancement courses
9. **MIL:** Modern Indian languages
10. **IKS:** Indian Knowledge System
11. **VEC:** Value Education Courses
12. **OJT:** On Job Training: (Internship/Apprenticeship)
13. **FP:** Field Projects
14. **CEP:** Community Engagement and Service
15. **CC:** Co-Curricular Courses
16. **RM:** Research Methodology
17. **RP:** Research Project/Dissertation

BCA First Year Semester I *(Level 4.5)*

Teaching Scheme

	Course Code	Course Name	Credits Assigned			Teaching Scheme (Hrs/ week)	
			Theory	Practical	Total	Theory	Practical
Optional 1	SBCACT1101	Problem Solving Techniques with C Lang.	02	--	**04**	02	--
	SBCACP1101	Problem Solving Techniques with C Lang. (P)	-	02			04
Optional 2	SBCAMT1101	Computer Architecture	02	--	**04**	02	--
	SBCAMP1101	Computer Architecture (P)	-	02			04
Optional 3	SBCAMT1101	Mathematics Foundations to Computer Science-I	02	--	**04**	02	--
	SBCAMP1101	Data Analysis with Excel(P)	-	02			04
Generic Electives *(from other Faculty)*	SBCAGE1101	Basics of Information Technology	02	--	**02**	02	--
Skill Based Course (related to Major)	SBCASC1101	Office Automation	--	02	**02**	--	04
Ability Enhancement Course	AECENG1101	L1 – Compulsory English	02	--	**02**	02	--
Indian Knowledge System (IKS)	IKSXXX1101	Select from **Basket 5**	02	--	**02**	02	--
Ability Enhancement Course (MIL)	**ACEMIL1101**		02	--	**02**	02	--
Total Credits			**14**	**08**	**22**	**12**	**20**

BCA First Year Semester I *(Level 4.5)*

Examination Scheme

[20% Continuous Assessment (***CA***) and 80% End Semester Assessment (***ESA***)]

(For illustration we have considered a paper of 02 credits, 50 marks, need to be modified depending on credits assigned to individual paper)

Subject (1)	Course Code (2)	Course Name (3)	Theory				Practical		Total Col (6+7) / Col (8+9) (10)
			Continuous Assessment (CA)			ESA			
			Test I (4)	Test II (5)	Average of T1 & T2 (6)	Total (7)	CA (8)	ESA (9)	
Optional 1	SBCACT1101	Problem Solving Techniques with C Lang.	10	10	10	40	--	--	50
	SBCACP1101	Problem Solving Techniques with C Lang. (P)	--	--	--	--	20	30	50
Optional 2	SBCAMT1101	Computer Architecture	10	10	10	40	--	--	50
	SBCAMP1101	Computer Architecture (P)	--	--	--	--	20	30	50
Optional 3	SBCAMT1101	Mathematics Foundations to Computer Science-I	10	10	10	40	--	--	50
	SBCAMP1101	Data Analysis with Excel(P)	--	--	--	--	20	30	50
Generic Elective	SBCAGE1101	Basics of Information Technology	10	10	10	40	--	--	50
Skill Based Course	SBCASC1101	Office Automation	--	--	--	--	20	30	50
Ability Enhancement Course	AECENG1101	**L1 – Compulsory English**	10	10	10	40	--	--	50
Indian Knowledge System (IKS)	IKSXXX1101	Select from **Basket 5**	10	10	10	40	--	--	50
Ability Enhancement Course (MIL)	**ACEMIL1101**		10	10	10	40	--	--	50

BCA First Year Semester II *(Level 4.5)*

Teaching Scheme

	Course Code	Course Name	Credits Assigned			Teaching Scheme (Hrs/ week)	
			Theory	Practical	Total	Theory	Practical
Optional 1	SBCACT1151	OOP's with Java	02	--	04	02	--
	SBCACP1151	OOP's with Java (P)	-	02			04
Optional 2	SBCAMT1151	Data Structures	02	--	04	02	--
	SBCAMP1151	Data Structures (P)	-	02			04
Optional 3	SBCAMT1151	Mathematics Foundations to Computer Science - II	02	--	04	02	--
	SBCAMP1151	R Language (P)	-	02			04
Generic Electives *(from other Faculty)*	SBCAGE1151	Logical Reasoning	02	--	02	02	--
Skill Based Course (related to Major)	SBCASC1151	Web Technologies	--	02	02	--	04
Ability Enhancement Course	AECENG1151	L1 – Compulsory English	02	--	02	02	--
Value Education Courses (VEC)	VECCOI1151	Constitution of India	02	--	02	02	--
Ability Enhancement Course (MIL)	**ACEMIL1151**		02	--	02	02	--
Total Credits			**14**	**08**	**22**	**12**	**20**

BCA First Year Semester II *(Level 4.5)*

Examination Scheme

[20% Continuous Assessment (***CA***) and 80% End Semester Assessment (***ESA***)]

(For illustration we have considered a paper of 02 credits, 50 marks, need to be modified depending on credits assigned to individual paper)

Subject (1)	Course Code (2)	Course Name (3)	Theory				Practical		Total Col (6+7) / Col (8+9) (10)
			Continuous Assessment (CA)			ESA			
			Test I (4)	Test II (5)	Average of T1 & T2 (6)	Total (7)	CA (8)	ESA (9)	
Optional 1	SBCACT1151	OOP's with Java	10	10	10	40	--	--	50
	SBCACP1151	OOP's with Java (P)	--	--	--	--	20	30	50
Optional 2	SBCAMT1151	Data Structures	10	10	10	40	--	--	50
	SBCAMP1151	Data Structures (P)	--	--	--	--	20	30	50
Optional 3	SBCAMT1151	Mathematics Foundations to Computer Science - II	10	10	10	40	--	--	50
	SBCAMP1151	R Language (P)	--	--	--	--	20	30	50
Generic Elective	SBCAGE1151	Logical Reasoning	10	10	10	40	--	--	50
Skill Based Course	SBCASC1151	Web Technologies	--	--	--	--	20	30	50
Ability Enhancement Course	AECENG1151	**L1 – Compulsory English**	10	10	10	40	--	--	50
Value Education Courses (VEC)	VECCOI1151	Constitution of India **Basket 5**	10	10	10	40	--	--	50
Ability Enhancement Course (MIL)	**ACEMIL1151**		10	10	10	40	--	--	50

Course Structure: *Major 1 -Teaching Scheme*

Course Code	Course Name (Paper Title)	Teaching Scheme(Hrs.)		Credits Assigned		
		Theory	Practical	Theory	Practical	Total
SBCACT1101	Problem Solving Techniques with C Lang	02	--	02	--	02

Major 1 -Assessment Scheme

Course Code (2)	Course Name (3)	Theory				Practical		Total [Col (6+7) or Col (8+9)] (10)
		CA			ESA (7)	CA (8)	ESA (9)	
		Test I (4)	Test II (5)	Avg. of T1 & T2 (6)				
SBCACT1101	Problem Solving Techniques with C Lang	10	10	10	40	--	--	50

SBCACT1101: *Problem Solving Techniques with C Lang (Major 1) Curriculum*

Course pre-requisite:

- This is an introductory programming course and hence no prerequisites

Course Objectives:

- CO1: Understand basic terminology of computers, problem solving, programming Languages and their evolution (Understand)
- CO2: Create specification from problem requirements by asking questions to disambiguate the requirement statement. (Create)
- 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). (Create)
- CO4: Translate an algorithm into a C computer program (Create)
- CO5: Testing and analysing programs using debugging tools. (Analyze)

Course Outcomes:

Students will be able to:

- Define algorithms and explain their characteristics
- Formulate algorithm and draw flow chart to solve a given problem
- Explain use of appropriate data types, control statements
- Demonstrate ability to use top-down program design

Details

Curriculum Details:*(There shall be FOUR Modules in each course)*

Course Content:

UNIT I: (CO-1, CO-2)

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.

UNIT II: (CO-2, CO-3, CO-4)

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.

UNIT III: (CO-2, CO-3, CO-4)
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.

UNIT IV: (CO-2, CO-3, CO-4, CO-5)
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.

Text Books

1. Venkatesh, Nagaraju Y, Practical C Programming for Problem Solving, Khanna Book Publishing Company, 2024.
2. AICTE's Programming for Problem Solving (with Lab Manual), Khanna Book Publishing Company, 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.

Course Structure: *Major 1 -Teaching Scheme*

Course Code	Course Name (Paper Title)	Teaching Scheme(Hrs.)		Credits Assigned		
		Theory	Practical	Theory	Practical	Total
SBCACP1101	Problem Solving Techniques with C Lang (practical)	--	02	--	02	02

Major 1 -Assessment Scheme

Course Code (2)	Course Name (3)	Theory				Practical		Total [Col (6+7) or Col (8+9)] (10)
		CA			ESA (7)			
		Test I (4)	Test II (5)	Avg. of T1 & T2 (6)		CA (8)	ESA (9)	
SBCACP1101	Problem Solving Techniques with C Lang (practical)	--	--	--	--	30	20	50

SBCACP1101: *Problem Solving Techniques with C Lang (practical) (Major 1)*

Problem Solving Techniques with C Lang: 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 of numbers 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 todisplay first 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?For each 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	Triangle		Diamond

pattern: ##### # # # # # # #####	Patterns with numbers: 1 121 12321 1234321 123454321	Square with diagonals: * * * * * * * * * * * * * *	Pattern * *** ***** *** *

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 are consecutive 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 the Taylor series for e^x?
14. Compute approximate value of sin(x)/cos(x) considering first n (n > 0) terms ofthe Taylor 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 3 2 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 ($0 \leq$ 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 5432 the output should be “FIVE FOUR THREE TWO”?
13. Using the grading scheme described in the question 4 (UNIT III), Compute how many students 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) in reading 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 are read using sentinel controlled repetition using only few variables?
15. Compute body mass index, BMI = weightinKGs / (HeightinMeters *HeightinMeters), 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 18.5 Normal:
    >=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 number is 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 and outputs mean (average), range (max-min) and mode (most frequent elements)?
4. Design a modular algorithm/program which reads an array of n integer elements and outputs median?
5. Implement your own string length and string reversal functions?
6. Design algorithm/program to perform matrix operations addition, subtractionand transpose?
7. Write a recursive program to count the number of digits of a positive integer number?
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 in reverse order?

******

Course Structure: *Minor 1 -Teaching Scheme*

Course Code	Course Name (Paper Title)	Teaching Scheme(Hrs.)		Credits Assigned		
		Theory	Practical	Theory	Practical	Total
SBCAMT1101	Computer Architecture	02	--	02	--	02

Minor 1 -Assessment Scheme

Course Code (2)	Course Name (3)	Theory				Practical		Total [Col (6+7) or Col (8+9)] (10)
		CA			ESA (7)			
		Test I (4)	Test II (5)	Avg. of T1 & T2 (6)		CA (8)	ESA (9)	
SBCAMT1101	Computer Architecture	10	10	40	10	--	--	50

SBCAMT1101: *Computer Architecture (Minor 1) Curriculum Details*

Course pre-requisite:

- This is an introductory programming course and hence no prerequisites

Course Objectives:

- 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.
- CO5: To Understand the concept of Parallel Processing.
- CO6: To understand the concept of memory organization.

Course Outcomes:

On completion of the course, student will be able to–

- Design of combinational circuits
- Design of sequential circuits
- Describe block diagram of CPU, Memory and types of I/O transfers

UNIT-I

Digital Principles: Definition for Digital signals, Digital logic, Digital computers, Von Neumann Architecture, Boolean Laws and Theorems, K-Map: Truth Tables to K-Map, 2, 3 and 4 variable K Map, K-Map Simplifications, Don't Care Conditions, SOP and POS.

Number Systems: Decimal, Binary, Octal, Hexadecimal, Number System Conversions, Binary Arithmetic, Addition and subtraction of BCD, Octal Arithmetic, Hexadecimal Arithmetic, Binary Codes, Decimal Codes, Error detecting and correcting codes, ASCII, EBCDIC, Excess- 3 Code, The Gray Code.

UNIT-II

Combinational Circuits: Half Adder and Full Adder, Subtractor, Decoders, Encoder, Multiplexer, Demultiplexer

Sequential Circuits: Flip-Flops- SR Flip- Flop, D Flip-Flop, J-K Flip-Flop, T Flip-Flop. **Register:** 4 bit register with parallel load, Shift Registers- Bidirectional shift register with parallel load Binary Counters-4 bit synchronous and Asynchronous binary counter.

UNIT-III

Basic Computer Organization and Design: Instruction Codes, Computer Registers, Computer Instructions, Timing and Control, Instruction Cycle, Memory-Reference Instructions, Input- Output Interrupt, Complete Computer Description, Design of Basic Computer, Design of Accumulator logic. Central Processing Unit: Introduction, General Register Organization, Stack Organization, Instruction Formats, Addressing Modes, Data Transfer and Manipulation, Program Control, Reduced Instruction Set Computer(RISC), RISC Vs CISC.

UNIT-IV

Pipeline and Vector Processing: Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline, RISC Pipeline. Input-Output Organization: Peripheral Devices, Input-Output Interface, Asynchronous data transfer, Modes of Transfer, Priority Interrupt, Direct memory Access, Input-Output Processor(IOP). Memory Organization: Memory Hierarchy, Main Memory, Auxiliary memory, Associate Memory, Cache Memory, Virtual Memory, Memory Management Hardware.

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, Sixth Edition,
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.

Course Structure: *Minor 1 -Teaching Scheme*

Course Code	Course Name (Paper Title)	Teaching Scheme(Hrs.)		Credits Assigned		
		Theory	Practical	Theory	Practical	Total
SBCAMP1101	Computer Architecture (practical)	--	02	--	02	02

Minor 1 -Assessment Scheme

Course Code (2)	Course Name (3)	Theory				Practical		Total [Col (6+7) or Col (8+9)] (10)
		CA			ESA (7)			
		Test I (4)	Test II (5)	Avg. of T1 & T2 (6)		CA (8)	ESA (9)	
SBCAMP1101	Computer Architecture (practical)	--	--	--	--	30	20	50

SBCAMP1101: *Computer Architecture (practical) (Minor 1)*

Suggestive Laboratory Experiments:

1. Verify logic behavior of AND, OR, NAND, NOR, EX-OR, EX-NOR, Invert and Buffer gates.
2. To study and verify NAND as a Universal Gate
3. To verify De- Morgan's theorem for 2 variables
4. Design and test of an S-R flip-flop using NAND/NOR gate.
5. Convert BCD to Excess-3 code using NAND gate
6. To Convert Binary to Grey Code
7. Verification of Truth Tables of J-K Flip-Flop using NAND/NOR gate
8. Realize Decoder and Encoder circuit using Basic Gates.
9. Design and implement the 4:1 MUX using gates.
10. Implementation of 4-Bit Parallel Adder Using 7483 IC.
11. Design and verify operation of half adder and full adder.
12. Design and verify operation of half subtractor.
13. Design and Implement a 4 bit shift register using Flip flops.
14. Implement Boolean function using logic gates in both SOP and POS
15. Design and Implement a 4 bit synchronous counter.
16. Design and verify 4 bit asynchronous counter.

Hardware

1. Familiarize the computer system layout: marking positions of SMPS, motherboard, FDD, HDD, CD, DVD and add on cards.
2. Identify the Computer Name and Hardware Specification (RAM capacity, Processor type, HDD, 32 bit/ 64 bit)
3. Identify and Troubleshoot the problems of RAM, SMPS and motherboard

4. Configure BIOS settings- disable and enable USB and LAN
5. Adding additional RAM to the system.(expanding RAM size).
6. To Study mother board layout of a system.
7. Demonstrate the assembly of a PC
8. Demonstration of various ports: CPU, VGA port, PS/2 (keyboard, mouse) ,USB, LAN, Speaker, Audio.
9. Install and configure windows OS
10. To study the installation of Printer and trouble shooting.

Course Structure: *Minor 2 -Teaching Scheme*

Course Code	Course Name (Paper Title)	Teaching Scheme(Hrs.)		Credits Assigned		
		Theory	Practical	Theory	Practical	Total
SBCAMT1102	Mathematics Foundations to Computer Science	02	--	02	--	02

Minor 2 -Assessment Scheme

Course Code (2)	Course Name (3)	Theory				Practical		Total [Col (6+7) or Col (8+9)] (10)
		CA			ESA (7)	CA (8)	ESA (9)	
		Test I (4)	Test II (5)	Avg. of T1 & T2 (6)				
SBCAMT1102	Mathematics Foundations to Computer Science	--	--	--	--	30	20	50

SBCAMT1102*: Mathematics Foundations to Computer Science (Minor 2) Curriculum Details*

Course pre-requisite:

- This is an introductory programming course and hence no prerequisites

Course Objectives:

- 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.

Course Outcomes:

On completion of the course, students will be able to–

- Relate and apply techniques for constructing mathematical proofs and make use
- of appropriate set operations, propositional logic to solve problems
- Use function or relation models to interpret associated relationships
- Apply basic counting techniques and use principles of probability
- Given a data, compute various statistical measures of central tendency
- Use appropriate Sampling techniques

Curriculum Details:*(There shall be FOUR Modules in each course)*

UNIT I: **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.

UNIT II:**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.

UNIT III:**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.

UNIT IV:**Matrix Algebra:**

Types of matrices, algebra of matrices–addition, subtraction, and multiplication of matrices, determinant of a matrix, symmetric and skew-symmetric matrices, orthogonal matrix, rank of a matrix, inverse of a matrix, applications of matrices to solve system of linear equations, Eigen values and Eigen vectors, Caley-Hamilton theorem.

Text Books

1. Garg, 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: An Applied 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

Course Structure: *Minor 2 -Teaching Scheme*

Course Code	Course Name (Paper Title)	Teaching Scheme(Hrs.)		Credits Assigned		
		Theory	Practical	Theory	Practical	Total
SBCAMP1102	Data Analysis with Excel(P)	--	04	--	02	02

Minot 2 -Assessment Scheme

Course Code (2)	Course Name (3)	Theory				Practical		Total [Col (6+7) or Col (8+9)] (10)
		CA			ESA (7)	CA (8)	ESA (9)	
		Test I (4)	Test II (5)	Avg. of T1 & T2 (6)				
SBCAMP1102	Data Analysis with Excel(P)	--	--	--	--	20	30	50

SBCAMP1102: Data Analysis with Excel(P) (Minor 2) Curriculum Details

Lab 1	
1.	New functions and features of Microsoft Excel
2.	Getting started with excel: Opening a blank or new workbook, general organization
3.	Highlights and main functions: Home, Insert, page layout, formulas
4.	Highlights and main functions: Data, review, view, add-inns
Lab 2	
5.	Customizing the Quick Access Toolbar,
6.	Working with Data: Entering, Editing, Copy, Cut, Paste, Paste Special
Lab 3	
7.	Manipulating Data, using Data Names and Ranges, Filters and Sortand Validation Lists
8.	Data from External Sources
9.	Basic Formulas and Use of Functions
10.	Data Analysis Using Charts and Graphs

Lab 4	
11.	Advanced Formulas and Functions, Advanced Worksheet Features
Lab 5	
12.	Advanced Data Analysis using PivotTables and Pivot Charts
Lab 6	
13.	Tabulation
14.	Bar diagram
15.	Multiple Bar diagram
16.	Pie diagram
Lab 7	
17.	Measure of central tendency: Mean, median, mode
Lab 8	
18.	Measure of dispersion: variance, standard deviation, Coefficient of variation
19.	Correlation, regression lines
Lab 9	
20.	t-test, F-test
Lab 10	
21.	ANOVA one way classification
Lab 11	
22.	t-test , F-test, ANOVA one way classification, chi square test, independence of attributes.
Lab 12	
23.	Chi square test, independence of attributes
Lab 13	
24.	Time series: forecasting Method of least squares
Lab 14	
25.	Moving average method, Inference and discussion of results.

Course Structure: *Generic Elective-Teaching Scheme*

Course Code	Course Name (Paper Title)	Teaching Scheme(Hrs.)		Credits Assigned		
		Theory	Practical	Theory	Practical	Total
SBCAGE1101	Basics of Info. Tech.	02	--	02	--	02

Generic Elective -Assessment Scheme

Course Code (2)	Course Name (3)	Theory				Practical		Total [Col (6+7) or Col (8+9)] (10)
		CA			ESA (7)			
		Test I (4)	Test II (5)	Avg. of T1 & T2 (6)		CA (8)	ESA (9)	
SBCAGE1101	Basics of Info. Tech.	10	10	10	40	--	--	50

SBCAGE1101: Basics of Computer Science (GE) Curriculum Details

Course pre-requisite:

1. Basic knowledge of computers

Course Objectives:

To Learn:

- The basic principles of computer.
- The basic level of Computer.

Course Outcomes:

Students will be able to:

- To learn Basic Function of Devices like I/O, HDD etc.
- To Understand the Fundamental of Software and Hardware. Understand the Concept of Operating System and Network.

Curriculum Details:*(There shall be FOUR Modules in each course)*

Module No.	Unit No.	Topic	Hrs. Required to cover the contents
1.0		**Introduction to Computer and History**	**7**
	1.1	Definition of Computer	
	1.2	Basic Computer Organization	
	1.3	Characteristics of Computer	
	1.4	Generations of Computer	
	1.5	Types of Computer: - Microcomputer, Minicomputer, Mainframe Computer, Workstations, Client and Server	
2.0		**Computer Peripherals & Memory**	**5**
	2.1	Input Devices :- Keyboard, Mouse, Trackball, Joystick, Light pen	
	2.2	Output Devices :- Monitor, Printer, Projector, Biometric Devices	
	2.3	Computer Memory :- RAM, ROM, Cache Memory	
3.0		**Storage Devices and Operating System**	**10**
	3.1	Compact Disk, Digital Versatile Disk	
	3.2	Hard Disk Drive	
	3.3	USB Flash Drive	
	3.4	Memory Card	
	3.5	Definition of operating System	
	3.6	Types of Operating System	
	3.7	Disk Operating System	

	3.8	Windows Operating System	
	3.9	Linux Operating System	
4.0		**Introduction to Computer Network & Internet**	
	4.1	Definition of Network	
	4.2	Types of Network :- LAN,MAN,WAN	
	4.3	Data Transmission Modes	
	4.4	OSI Model	**8**
	4.5	E-Mail	
	4.6	File Transfer Protocol	
	4.7	Web Browser	
	4.8	Types of Web Browser	
		Total	**30**

Reference Books:

1. Fundamental of Computer –5th& 6th Edition, P.K.Sinha, BPB Publication
2. Fundamental of Computer - V. Raja Raman, PHI Publication

Course Structure: *Skill based course -Teaching Scheme*

CourseCode (2)	Course Name (3)	Theory				Practical		Total [Col (6+7) or Col (8+9)] (10)
		CA			ESA (7)			
		Test I (4)	Test II (5)	Avg. of T1 & T2 (6)		CA (8)	ESA (9)	
SBCASC1101	Office Automation	--	--	--	--	25	25	50

Skill based course -Assessment Scheme

Course Code	Course Name (Paper Title)	Teaching Scheme(Hrs.)		Credits Assigned		
		Theory	Practical	Theory	Practical	Total
SBCASC1101	Office Automation	--	02	--	02	02

SBCASC1101: ***Office Automation (Skill based course) Curriculum Details***

1) Study of Word Opening screen
2) Study of EXCEL Opening screen
3) Study of PowerPoint Opening screen
4) Study of Access Opening screen
5) Study of Find and Replace Dialog Box in Microsoft Word
6) Study of Custom Dictionary & Go to Dialog Box
7) Study of Table Formatting
8) Study of mail merge
9) Study of creating charts.
10) Study of border and shading dialog box
11) Study of paragraph dialog box
12) Working of Formulas in Excel
13) Creating Presentation in Power Point
14) Creating database file in Access

Course Structure: *Major 1 -Teaching Scheme*

Course Code	Course Name (Paper Title)	Teaching Scheme(Hrs.)		Credits Assigned		
		Theory	Practical	Theory	Practical	Total
SBCACT1151	OOPS with Java	02	--	02	--	02

Major 1 -Assessment Scheme

Course Code (2)	Course Name (3)	Theory				Practical		Total [Col (6+7) or Col (8+9)] (10)
		CA			ESA (7)			
		Test I (4)	Test II (5)	Avg. of T1 & T2 (6)		CA (8)	ESA (9)	
SBCACT1151	OOPS with Java	10	10	10	40	--	--	50

SBCACT1151: *OOPS with JAVA (Major 1) Curriculum Details*

Course pre-requisite:

- Knowledge of Problem Solving Techniques using C programming language.

Course Objectives:

- 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

Course Outcomes:

Students will be able to:

- Use the syntax and semantics of java programming language and basic concepts of OOP.
- Develop reusable programs using the concepts of inheritance, polymorphism, interfaces and packages
- Apply the concepts of Exception handling to develop efficient and error free codes.

Curriculum Details:*(There shall be FOUR Modules in each course)*

UNIT I:

Fundamentals of Object Oriented Programming: Basic Concepts of Object Oriented Programming (OOP), Benefits and Applications of OOP.

Java Evolution: Java Features, Difference between Java, C and C++, Java and Internet, Java Environment.

Overview of Java Language: Introduction to Simple Java Program, Use of Comments and Math function, Application of two classes, Java Program Structure, Java Tokens and statements, Implementing Java program and JVM, Command Line Arguments.

Text Book 1: Chapters 1, 2 and 3.

UNIT II:

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.

Text Book 1: Chapters 4, 5, 6, and 7.

UNIT III:

Classes, Objects and Methods: Defining Class, Methods Declaration, Constructors, Methods Overloading, Overriding Methods, Inheritance

Arrays, Strings and Vectors: 1D arrays, Creating an Array, 2D arrays, Strings, Vectors, Wrapper Classes, Enumerated Types

Inheritance: Defining, extending classes, and Implementing Interfaces. Multiple inheritance and polymorphism.

Text Book 1: Chapters 8, 9, and 10.

UNIT IV:

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

Text Book 1: Chapters 11 & 13.

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 Private Limited, 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, OxfordUniversity Press, 2014.

Web Resources

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

Course Structure: *Major 1 -Teaching Scheme*

Major 1 -Assessment Scheme

Course Code	Course Name (Paper Title)	Teaching Scheme(Hrs.)		Credits Assigned		
		Theory	Practical	Theory	Practical	Total
SBCACP1151	OOPS with Java	--	04	--	02	02

Course Code (2)	Course Name (3)	Theory				Practical		Total [Col (6+7) or Col (8+9)] (10)
		CA			ESA (7)			
		Test I (4)	Test II (5)	Avg. of T1 & T2 (6)		CA (8)	ESA (9)	
SBCACP1151	OOPS with Java	--	--	--	--	20	30	50

SBCACP1151: *OOPS with JAVA (Major 1) Curriculum Details*

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 and print 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 non-recursive 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 a palindrome.
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 with methods to add and subtract. Create another class calledMulDiv 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.
    a. USN, NAME, BRANCH, PHONE, PERCENTAGE
    b. 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 words in a text.
13. Write a Java program to create a class called Shape with methods called getPerimeter() and getArea(). Create a subclass called Circle that overrides the getPerimeter() and getArea()

methods to calculate the area and perimeter of a circle.

14. Write a Java program to create a class Employee with a method called calculateSalary(). Create two subclasses Manager and Programmer. In each subclass, 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 additional member variables, constructors also in classes 'SBI' and 'PNB'.
16. Write a Java package program for the class book and then import the data from the package and display the result.
17. Write a Java program for finding the cube of a number using a package for various data types and then import it in another class and display the results.
18. Write a Java program for demonstrating the divide by zero exception handling.
19. Write a Java program that reads a list of integers from the user and throws an exception 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 Structure: *Minor 1 -Teaching Scheme*

Minor 1 -Assessment Scheme

		Theory	Practical	

Course Code (2)	Course Name (3)	CA			ESA (7)			Total [Col (6+7) or Col (8+9)] (10)
		Test I (4)	Test II (5)	Avg of T1 & T2 (6)	CA (8)		ESA (9)	
	(Paper Title)	Teaching Scheme(Hrs.)			Credits Assigned			
		Theory	Practical		Theory	Practical		
SBCAMT1151	Data Structures	10	10	10	40	--	--	50
SBCAMT1151	Data Structures	02	--		02		--	02

SBCAMT1151: *Data Structures (Minor 1) Curriculum Details*

Course pre-requisite:

- Programming Fundamentals: Understanding the basic syntax and semantics of C programming language.
- Problem-Solving Skills: Ability to break down a problem into smaller steps and devise a step-by-step solution and familiarity with simple algorithms

Course Objectives:

- 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.

Course Outcomes:

- Understand basic data structure.
- Create and use various data structures like Strings, Arrays, Linked Lists, and Trees.

Curriculum Details:*(There shall be FOUR Modules in each course)*

UNIT I:

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.

UNIT II:

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.

UNIT III:

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.

UNIT IV:

Graphs: Definition, Terminology, Representation, Traversal.

Trees: Definition, Terminology, Binary Trees, Traversal of Binary Tree, Binary Search Tree, Inserting, Deleting and Searching in Binary Search Tree, Height Balanced Trees: AVL Trees, Insertion and Deletion in AVL Tree.

Text Books

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

Reference Books

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

Web Resources

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

Course Code	Course Name (Paper Title)	Teaching Scheme(Hrs.)		Credits Assigned		
		Theory	Practical	Theory	Practical	Total
SBCAMP1151	Data Structures (P)	--	04	--	02	02

Course Structure: *Minor 1 -Teaching Scheme*

Minor 1 -Assessment Scheme

Course Code (2)	Course Name (3)	Theory				Practical		Total [Col (6+7) or Col (8+9)] (10)
		CA			ESA (7)			
		Test I (4)	Test II (5)	Avg of T1 & T2 (6)		CA (8)	ESA (9)	
SBCAMP1151	Data Structures (P)	--	--	--	--	20	30	50

SBCAMP1151: *Data Structures (P) (Minor 1) Curriculum Details*

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 Binary Search.
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.

Note: - Conduct at least ten practical on given contents.

Course Structure: *Minor 2 -Teaching Scheme*

Course Code	Course Name (Paper Title)	Teaching Scheme(Hrs.)		Credits Assigned		
		Theory	Practical	Theory	Practical	Total
SBCAMT1152	Mathematics Foundations to Computer Science - II	02	--	02	--	02

Minor 2 -Assessment Scheme

Course Code (2)	Course Name (3)	Theory				Practical		Total [Col (6+7) or Col (8+9)] (10)
		CA			ESA (7)			
		Test I (4)	Test II (5)	Avg. of T1 & T2 (6)		CA (8)	ESA (9)	
SBCAMT1152	Mathematics Foundations to Computer Science - II	10	10	10	40	--	--	50

SBCAMT1152: *Mathematics Foundations to Computer Science - II (Minor 2)*

Course pre-requisite:

- Familiarity with basic concepts such as set theory, logic, relations, and functions.
- Prior exposure to topics like graph theory, and number theory.

Course Objectives:

- 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.

Course Outcomes:

Students will be able to:

- Understand and apply correct methods of reasoning, constructing formal proofs, and logical arguments in computational problems.
- Utilize foundational mathematical techniques for advanced computational methods, including numerical methods and optimization, to solve complex problems.
- Develop and implement various problem-solving strategies, enabling students to address both theoretical and practical challenges in computer science.

Curriculum Details:*(There shall be FOUR Modules in each course)*

Course Content:

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.

UNIT II:

Algebraic Structures:

Semi-group, Monoid, Group, Subgroup, Cyclic group.

UNIT III:

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.

UNIT IV:

Optimization Techniques:
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.

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, Pearson Prentice Hall, 2003.
4. S.B. Singh, Discrete Structures, Khanna Book Publishing, 2023 (AICTE Recommended Textbook)

Reference Books

1. Rosen Kenneth H. and Krithivasan Kamala, Discrete Mathematics

and its Applications, McGraw Hill, India, 2019.
2. Chakravorty J. G. and Ghosh P. R., Linear Programming and Game Theory, Moulik Library, 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 Structure: *Minor 2 -Teaching Scheme*

Course Code	Course Name (Paper Title)	Teaching Scheme(Hrs.)		Credits Assigned		
		Theory	Practical	Theory	Practical	Total
SBCAMP1152	R Language (P)	--	04	--	02	02

Minor 2 -Assessment Scheme

Course Code (2)	Course Name (3)	Theory				Practical		Total [Col (6+7) or Col (8+9)] (10)
		CA			ESA (7)			
		Test I (4)	Test II (5)	Avg. of T1 & T2 (6)		CA (8)	ESA (9)	
SBCAMP1152	R Language (P)	--	--	--	--	20	30	50

SBCAMP1152: *R Language (Minor 2) Curriculum Details*

List of experiments

1. Write a R program to create a list containing strings, numbers, vectors and a logical values.
2. Write a R program to list containing a vector, a matrix and a list and give names to the elements in the list.
3. Write a R program to create a list containing a vector, a matrix and a list and give names to the elements in the list. Access the first and second element of the list.
4. Write a R program to create a list containing a vector, a matrix and a list and add element at the end of the list.
5. Write a R program to select second element of a given nested list.
6. Write a R program to create a list containing a vector, a matrix and a list and remove the second element.

7. Write a R program to create a list containing a vector, a matrix and a list and update the last element.
8. Write a R program to merge two given lists into one list.
9. Write a R program to convert a given list to vector.
10. Write a R program to create a list of dataframes and access each of those dataframes from the list.
11. Write a R program to count number of objects in a given list.
12. Write a R program to convert a given dataframe to a list by rows.
13. Write a R program to convert a given matrix to a list.
14. Write a R program to assign NULL to a given list element.
16. Write a R program to Add 10 to each element of the first vector in a given list

17. Write a R program to extract all elements except the third element of the first vector of a given list.
18. Write a R program to add a new item g4 = "Python" to a given list.
19. Write a R program to assign new names "a", "b" and "c" to the elements of a given list.
20. Write a R program to get the length of the first two vectors of a given list.
21. Write a R program to find all elements of a given list that are not in another given list.
22. Program based on Numerical Interpolation
23 program based on Numerical Integration
24 Program based on Linear programming
25 Program based on Transportation problem

Course Structure: *Generic Electives -Teaching Scheme*

Generic Electives -Assessment Scheme

Course Code (2)	Course Name (3)	Theory				Practical		Total [Col (6+7) or Col (8+9)] (10)
		CA			ESA (7)			
		Test I (4)	Test II (5)	Avg. of T1 & T2 (6)		CA (8)	ESA (9)	
SBCAGE1151	Logical Reasoning	10	10	10	40	--	--	50

SBCAGE1151: *Logical Reasoning (Generic Electives) Curriculum Details*

Course pre-requisite:

1. Basic knowledge of English
2. Basic knowledge of Numbers.
3. Basic knowledge of general knowledge.

Course Objectives:

1 This course enables students to develop their ability to reason by introducing them to elements of reasoning
2 Basics knowledge of different types of Series
3 Study of Coding and Decoding
4 Knowledge of Blood Relations, Directions and Puzzles

Course Outcomes:

1. Develops ability to think logically of student
2. Understanding Relations, Directions, Arrangements, Logics, Puzzles.
3. Improves Mental Alertness
4. Construct a logically sound and well-reasoned argument

SBCAGE1151: *Logical Reasoning (Generic Electives) Curriculum Details*

Module No.	Unit No.	Topic	Hrs. Required to cover the contents
1.0		**Series, Analogy**	**8**
	1.1	Types of series	
	1.2	Alphabet series	
	1.3	Alpha numeric	
	1.4	Completing the Analogous Pair	
	1.5	Direct/Simple Analogy	
	1.6	Choosing the Analogous Pair	
	1.7	Double Analogy	
	1.8	Number analogy	
2.0		**Direction Sense Test**	**8**
	2.1	Problems based on angular changes in direction	
	2.2	Problems on Shadows	
	2.3	General Problems based on Pythagoras Theorem	
3.0		**Coding-Decoding**	**7**
	3.1	Letter coding	
	3.2	Direct Letter Coding	
	3.3	Number/Symbol Coding	
4.0		**Blood Relation**	**7**
	4.1	Concepts of deciphering relations based problems	
	4.2	Problems on deciphering jumbled up descriptions	
	4.3	Relation puzzle	
	4.3	Coded relations.	
		Total	**30**

Reference Books:

1. Modern Approach to Verbal &Dr.R.S Aggarwal S.Chand and Company
   NonVerbal Reasoning

2. Test of Reasoning Edgar Thorpe McGraw Hill Education

Course Code	Course Name (Paper Title)	Teaching Scheme(Hrs.)		Credits Assigned		
		Theory	Practical	Theory	Practical	Total
SBCASC1151	Web Technologies	--	02	--	02	02

Course Structure: *Skill Based Course -Teaching Scheme*

Skill Based Course -Assessment Scheme

Course Code (2)	Course Name (3)	Theory				Practical		Total [Col (6+7) or Col (8+9)] (10)
		CA			ESA (7)			
		Test I (4)	Test II (5)	Avg of T1 & T2 (6)		CA (8)	ESA (9)	
SBCASC1151	Web Technologies	--	-	--	--	20	30	50

SBCASC1151: *Web Technologies (Skill Based Course) Curriculum Details*

Course pre-requisite:

1. Proficiency in at least one programming language, such as Python, Java, or C++. Understanding of programming concepts such as loops, conditionals, functions, and data structures like arrays, lists.
2. Familiarity with object-oriented programming (OOP) principles, including classes, objects, inheritance, and polymorphism.

Course Objectives:

CO1: To understand the concepts and architecture of the World Wide Web, Markup languages along with Cascading Style Sheets.
CO2: To understand the concepts of event handling and data validation mechanisms.
CO3: To understand the concepts of embedded dynamic scripting on client and server side Internet Programming and basic full stack web development.
CO4: To develop modern interactive web applications

Course Outcomes:

Student will able to:

- Demonstrate basics concepts of web technologies.
- Work with CSS
- Work with data validation

Curriculum Details:*(There shall be FOUR Modules in each course)*

Unit I:

Introduction to HTML, history of HTML, Objective, basic Structures of HTML, Header Tags, body tags, Paragraph Tags.
Tags for FORM Creation, TABLE, FORM, TEXTAREA, SELECT, IMG, IFRAME FIELDSET, ANCHOR.
Lists in HTML, Introduction to DIV tag, NAVBAR Design.
Introduction to CSS, types, Selectors, and Responsiveness of a web page.
Introduction to Bootstrap, downloads/linking, using classes of Bootstrap, understanding the Grid System in Bootstrap.
Introduction to www, Protocols and Programs, Applications and development tools, web browsers, DNS, Web hosting Provider, Setting up of Windows/Linux/Unix web servers, Web hosting in cloud, Types of Web Hosting.

Unit II:

Introduction to JavaScript: Functions and Events, Document Object model traversing using JavaScript. Output System in JavaScript i.e. Alert, throughput, Input box, Console. Variables and Arrays in JavaScript.Date and String handling in JavaScript.
Manipulating CSS through JavaScript: Form Validation like Required validator, length validator, Pattern validator. Advanced JavaScript, Combining HTML, CSS and JavaScript events and buttons, controlling your browser.Introduction to AJAX, Purpose, advantages and disadvantages, AJAX based Web applications and alternatives of AJAX.
Introduction to XML: uses, Key concepts, DTD 8 schemas, XSL, XSLT,and XSL Elements and transforming with XSLT. Introduction to XHTML.
JSON: Introduction to JSON, Keys and Values, Types of Values, Arrays, Objects

Text Books

1. Laura Lemay, Mastering HTML, CSS & Java Script Web Publishing, BPB Publications, 2016
2. Thomas A. Powell, The Complete Reference HTML & CSS, Fifth Edition, 2017

Reference Books

1. Silvio Moreto, Bootstrap 4 By Example, ebook, 2016.
2. Tanweer Alam, Web Technologies, Khanna Book Publishing, 2011.

Web Resources

1. www.javatpoint.com
2. www.w3schools.com
3. https://www.geeksforgeeks.org/web-technology/

Practical list of Programs:

PART-A

1. Create your class time table using table tag.
2. Design a Webpage for your college containing description of courses, department, faculties, library etc. using list tags, href tags, and anchor tags.

3. Create web page using Frame with rows and columns where we will have header frame, left frame, right frame, and status bar frame. On clicking in the left frame, information should be displayed in right frame.
4. Create Your Resume using HTML, use text, link, size, color and lists.
5. Create a Web Page of a super market using (internal CSS)
6. Use Inline CSS to format your resume that you have created.
7. Use External CSS to format your time table created.
8. Use all the CSS (inline, internal andexternal) toformat college web page that you have created.
9. Write a HTML Program to create your college website using for mobile device.

PART – B

1) Write an HTML/JavaScript page to create login page with validations.
2) Develop a Simple calculator for addiction, subtraction, multiplication and division operation using JavaScript.
3) Use Regular Expressions for validations in Login Page using JavaScript.
4) Write a Program to retrieve date from a text file and displaying it using AJAX.
5) Create XML file to store Student Information like Register Number, Name, Mobile Number, DOB, and Email-Id.
6) Create a DTD for (0).
7) Create XML scheme for (0).
8) Create XSL file to convert XML file to XHTML file.
9) Write a JavaScript program using Switch case.
10) Write a JavaScript program using any 5 events.
11) Write a JavaScript program using built in JavaScript objects.
12) Write program for populating values from JSON text.
13) Write a program to transform JSON text to a JavaScript object.

Guidelines for the Course Assessment:

A. Continuous Assessment (CA) (20% of the Maximum Marks) of theory and practical courses:

i. **For Theory Course:** CA shall form 20% of the Maximum Marks and shall be carried out over the entire semester. It shall be done by conducting **Two Tests** (Test I on 40% curriculum) and **Test II** (on remaining 40% syllabus) and average of the marks scored by a student in these two tests of a particular paper shall be taken as the **CA** score.

ii. **For Practical Course:** CA score of the practical course shall be marks scored by a student in the internal practical examination conducted by the concerned teacher.

B. End Semester Assessment (80% of the Maximum Marks) of theory and practical courses:

(For illustration a paper of 02 credits, 50 marks has been considered and shall be modified appropriately depending upon credits of the individual paper)

Question Paper Pattern of the ESA:

i. **ESA Question paper shall consist 6 questions, each of 10 marks**
ii. **Question No.1 shall be compulsory and shall be based on the entire syllabus**
iii. Students shall have to solve ***ANY THREE*** of the remaining Five Questions (i.e. from question 2 to 6)
iv. **Students shall have to solve a TOTAL of 4 Questions.**

C. Assessment of On Job Training (OJT) Course (for 04 credits):

a. Continuous assessment part (**40%, 40 marks out of 100**) of this course shall be done by the mentor of the student, where he /she is supposed to complete his On Job Training. This shall be based on the regularity, participation and performance of the students at the place of OJT.

b. Semester End Assessment (ESA) (**60% of the total marks, 60 marks out of 100**) of this course shall be done by a panel of examiners in two parts

i. based on the work report submitted by the student (**50% i.e. 30 marks)** and

ii. **Remaining 50%** (30 marks) shall be based on his presentation and viva-voce on the work carried to be assessed by the panel of examiners. This assessment shall be done along with practical examinations of respective courses / subjects.

D. Assessment of Field Project (FP) and Research Project (RP) (e.g. for 02 credits)

a. Continuous assessment part (**40%, 20 marks out of 50**) of this course shall be done by the mentor of the student and shall be based on regularity, experimental work and performance of the student.

b. Semester End Assessment (ESA) (**60% of the total marks, 30 marks out of 50**) of this course shall be done shall be done by a panel of examiners in two parts

i. based on the work report submitted by the student (**50% i.e. 15 marks)** and

ii. **Remaining 50%** (15 marks) shall be based on his presentation and viva-voce on the work carried out by the student. This assessment shall be done along with practical examinations of the respective courses / subjects.

E. Assessment of Co-Curricular courses (CCC):

a. Assessment of the CCC course shall be done by the respective course coordinator as a part of CA and be based on the regularity, performance of a student and his participation in various activities as prescribed in the regulations prepared in this regard.

b. The End Semester Assessment (ESA) of the CCC courses shall be done as per the regulations prepared in this regard and shall be done on the basis of the write-up, presentation by the student on the activities that he has carried out in a semester.

c. Students shall have freedom to opt for more than one CCC courses. However, score of the best performing CC shall be considered for preparing his result.

F. Syllabi, Teaching and Examination Scheme for the courses in Column 7 and Column 8 (AEC, VEC, IKS, CI, EVS, CCCs, etc.) shall be common for all the students from different faculties.

Note: Number of lectures required to cover syllabus of a course depends on the number of credits assigned to a particular course. One credit of theory corresponds to 15 Hours lecturing and for practical course one credit corresponds to 30 Hours. For example, for a course of two credits 30 lectures of one hour duration are assigned, while that for a three credit course45 lectures.

%%%%%%%%