K. C. E. Society's
Moolji Jaitha College
An 'Autonomous College' Affiliated to
K.B.C. North Maharashtra University, Jalgaon.
NAAC Reaccredited Grade - A (CGPA : 3.15 - 3rd Cycle)
UGC honoured "College of Excellence" (2014-2019)
DST(FIST) Assisted College

KNOWLEDGE IS POWER
ESTD. 1945

के. सी. ई. सोसायटीचे
मूळजी जेठा महाविद्यालय
क.व.चौ. उत्तर महाराष्ट्र विद्यापीठ, जळगाव संलग्नित
'स्वायत्त महाविद्यालय'
नॅकद्वारा पुनर्मानांकित श्रेणी –'ए' (सी.जी.पी.ए. : ३.१५ – तिसरी फेरी)
विद्यापीठ अनुदान आयोगाद्वारा घोषित 'कॉलेज ऑफ एक्सलन्स' (२०१४–२०१९)
डी.एस.टी. (फीस्ट) अंतर्गत अर्थसहाय्य प्राप्त

Date :- 01/08/2024

NOTIFICATION

Sub :- CBCS Syllabi of BCA (Sem. I & II)

Ref. :- Decision of the Academic Council at its meeting held on 27/07/2024.

The Syllabi of BCA (First and Second Semesters) as per **NATIONAL EDUCATION POLICY – 2020 (2024 Pattern)** and approved by the Academic Council as referred above are hereby notified for implementation with effect from the academic year 2024-25.

Copy of the Syllabi Shall be downloaded from the College Website (www.kcesmjcollege.in)

Sd/-
Chairman,
Board of Studies

To :

1) The Head of the Dept., M. J. College, Jalgaon.
2) The office of the COE, M. J. College, Jalgaon.
3) The office of the Registrar, M. J. College, Jalgaon.

Khandesh College Education Society's

Moolji Jaitha College, Jalgaon

An "Autonomous College"

Affiliated to

KavayitriBahinabai Chaudhari
North Maharashtra University, Jalgaon-425001

KNOWLEDGE IS POWER
ESTD. 1945

STRUCTURE

B.C.A. Honours / Honours with Research

as per AICTE Guidelines

[w.e.f. Academic Year: 2024-25]

Credit distribution structure for Three/ Four year Honors/ Honors with Research Degree Programme with Multiple Entry and Exit

Level 1	Sem	Core Courses	AEC	MDE	VAC	SEC	DSE	Cumulative Credits/ Sem	Degree/ Cumulative Cr.
4.5	I	2T 2P 2T 2P 2T 2P	4T	2T	2T	-	-	20	40
	II	2T 2P 2T 2P 2T 2P	2T	2T	2T	2T	-	20	
	Cum. Cr.	24	6	4	4	2	-	40	
Exit option: Award of UG Certificate with 40 credits and an additional 4 credits core NSQF course/ Internship OR Continue with the course.									

Sem- Semester, **AEC**- Ability Enhancement Course **MDE**- Multi-disciplinary Elective Course, **VAC**- Value Added Course, **SEC**- Skill Enhancement Course,**DSE**- Department Specific Elective, **T**- Theory, **P**- Practical.

Level 2	Sem	Core Courses	AEC	MDE	VAC	SEC	DSE	Cumulative Credits/ Sem	Degree/ Cumulative Cr.
5.0	III	2T 2P 2T 2P 2T 2P		2T	2P	2T 2P		20	80
	IV	2T 2P 2T 2P 2T 2P 2T			2T	2T2P		20	
	Cum. Cr.	24+ 26	6	4+2	4+4	2+8	-	40+40	

Level3	Sem	Core Courses	AEC	MDE	VAC	SEC	DSE / Audit course	Cumulative Credits/ Sem	Degree/ Cumulative Cr.
5.5	V	2T 2P 2T 2P				4P 0P	2T 2P 2T 2P 0T	20	120
	VI	2T 2P 2T				2T 4P	2T 2P 2T 2P 0T	20	
	Cum. Cr.	24+ 26+14	6	4+2	4+4	2+8+10	16	40+40+40	

B.C.A. Honours

Level4	Sem	Core Courses	AEC	MDE	VAC	SEC	OE	DSE / Audit course	Cumulative Credits/ Sem	Degree/ Cumulative Cr.
5.5	VII	2T 2P				0P 4P	4T	2T 2P 2T 2P	20	160
	VIII					8P		2T 2P 2T 2P 2T 2P	20	
	Cum. Cr.	24+ 26+14+ 4	6	4+2	4+4	2+8+10+12	4	16+20	40+40+40+40	

B.C.A. Honours with Research

Level 4	Sem	Core Courses	AEC	MDE	VAC	SEC	OE	DSE / Audit course	Cumulative Credits/ Sem	Degree/ Cumulative Cr.
5.5	VII	2T 2P 2T 2P				4RP		2T 2P 2T 2P	20	160
	VIII					20RP			20	
	Cum. Cr.	24+ 26+14+ 8	6	4+2	4+4	2+8+10+24		16+8	40+40+40+40	

Sem- Semester, **DSC**- Department Specific Course, **DSE**- Department Specific Elective, **T**- Theory, **P**- Practical, CC- Cocurricular**RM**- Research Methodology, **OJT**- On Job Training, **FP**- Field Project, **Int**- Internship, **RP**- Research Project,

FYBCA Structure and Syllabus (A.Y. 2024-25)

Sem	Course / Module	Hours / week	Credit	L	T	P	Code	Title
I	**3 WEEKS COMPULSORY INDUCTION PROGRAM (UHV-I)**							
	CC	2	2	1	1	0	BCA-CC-111	Mathematics Foundations to Computer Science - I
	CC	4	4	3	1	0	BCA-CC-112	Computer Architecture
	CC	4	2	0	0	4	BCA-CC-113	Practical on Computer Architecture
	SEC	2	2	1	1	0	BCA-SEC-111	Problem Solving Techniques
	SEC	4	2	0	0	4	BCA-SEC-112	Practical on Problem Solving Techniques
	AEC	4	4	3	1	0	BCA-AEC-111	Business Communication - I
	MDE	2	2	2	0	0	IKS-111	Indian Knowledge System
	VAC	2	2	2	0	0	ES-VEC-111	Environmental Studies
	Total Credits		**20**					

Sem	Course / Module	Hours / week	Credit	L	T	P	Code	
II	CC	2	2	1	1	0	BCA-CC-121	Mathematics Foundations to Computer Science - II
	CC	4	4	3	1	0	BCA-CC-122	Data Structures
	CC	4	2	0	0	4	BCA-CC-123	Practical on Data Structures
	CC	4	4	3	1	0	BCA-CC-124	Operating Systems
	CC	4	2	0	0	4	BCA-CC-125	Practical on Operating Systems
	SEC	2	2	1	1	0	BCA-SEC-121	Object Oriented Programming using Java
	SEC	4	2	0	0	4	BCA-SEC-122	Practical on Object Oriented Programming using Java
	VAC	2	2	2	0	0	CI-VEC-121	Indian Constitution

After Year 1, Students are advised to take Social Responsibility & Community Engagement - encompassing Community Engagement with an NGO in the vacation time.

An UNDER GRADUATE CERTIFICATE IN COMPUTER APPLICATION will be awarded, if a student wishes to exit at the end of First year.

Exit Criteria after First Year of BCA Programme

Students will have the option to exit the Bachelor of Computer Application (BCA) program after successfully completing the first year. Upon exit, they will be awarded a **UG Certificate in Computer Application**. To be eligible for this certificate, students must complete an additional 04 credits in one of the following areas:

1. **Skill-Based Subject**: A course designed to enhance practical and technical skills in the field of computer applications.

2. **Work-Based Vocational Course**: A vocational course offered during the summer term that emphasizes hands-on training and workplace readiness.

3. **Internship/Apprenticeship**: A professional internship or apprenticeship program in a relevant field, with a minimum duration of 08 weeks, which will take place after the second semester.

4. **Social Responsibility & Community Engagement**: Active engagement with an NGO or community organization for a minimum duration of 08 weeks, focusing on real-world problem-solving, social responsibility, and community service.

The exiting students will clear the subject / submit the Internship Report as per the College Schedule

Re-entry Criteria in to Second Year (Third Semester)

The student who takes an exit after one year with an award of certificate may be allowed to re-enter in to Third Semester for completion of the BCA Program after earning requisite credits in the First year.

Exam Pattern: Each theory and practical course of 4 credits will be of 100 marks comprising of 40 marks internal and 60 marks external examination. in case of courses of 2 credits, each theory and practical course will be of 50 marks comprising of 20 marks internal and 30 marks external examination

Rules of Continuous Internal Evaluation:

The Continuous Internal Evaluation for theory papers shall consist of two methods:

1. **Continuous & Comprehensive Evaluation (CCE):**

CCE will carry a maximum of 30% weightage (30/15 marks) of the total marks for a course. Before the start of the academic session in each semester, the subject teacher should choose any three assessment methods from the following list, with each method carrying 10/5 marks:

i. Individual Assignments
ii. Seminars/Classroom Presentations/Quizzes
iii. Group Discussions/Class Discussion/Group Assignments
iv. Case studies/Case lets
v. Participatory & Industry-Integrated Learning/Field visits
vi. Practical activities/Problem Solving Exercises
vii. Participation in Seminars/Academic Events/Symposia, etc.
viii. Mini Projects/Capstone Projects
ix. Book review/Article review/Article preparation
x. Any other academic activity

Each chosen CCE method shall be based on a particular unit of the syllabus, ensuring that three units of the syllabus are mapped to the CCEs.

2. Internal Assessment Tests (IAT):

IAT will carry a maximum of 10% weightage (10/5 marks) of the total marks for a course. IAT shall be conducted at the end of the semester and will assess the remaining unit of the syllabus that was not covered by the CCEs. The subject teacher is at liberty to decide which units are to be assessed using CCEs and which unit is to be assessed on the basis of IAT.

The overall weightage of Continuous Internal Evaluation (CCE + IAT) shall be 40% of the total marks for the course. The remaining 60% of the marks shall be allocated to the semester-end examinations.

The subject teachers shall communicate the chosen CCE methods and the corresponding syllabus units to the students at the beginning of the semester to ensure clarity and proper preparation.

SEMESTER -I

BCA-CC-111 Mathematics Foundation to Computer Science – I

Total Hours: 30 **Credits: 2**

Course objectives	• Understand the fundamental mathematical concepts such as sets, relations, functions, and their application to computer science. • Learn counting techniques and recurrence relations to solve computational problems efficiently. • Familiarize students with elementary graph theory and its applications in areas such as networking and optimization. • Introduce matrix algebra and its role in solving linear equations and other computer science-related problems.	
Course outcomes	• Apply set theory and functions to model and analyse computational problems. • Use combinatorial techniques and recurrence relations to solve complex problems like Fibonacci sequences and the tower of Hanoi. • Explore graph theory concepts such as Euler and Hamiltonian graphs and their application to optimization and network theory. • Utilize matrix algebra to solve linear systems, find eigenvalues and eigenvectors, and analyse problems in computer science.	
Unit	**Content**	**Hours**
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.	**7**
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.	**8**
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	**7**

	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	**8**
Study Resources	**Text Books** • Garg, Reena, Engineering Mathematics, Khanna Book Publishing Company, 2024. (AICTE Recommended Textbook) • Garg, Reena, Advanced Engineering Mathematics, Khanna Book Publishing Company, 2023. Model curriculum for UG Degree in BCA 30 • Kolman B., Busby R. and Ross S., Discrete Mathematical Structures, 6th Edition, Pearson Education, 2015. • Deo Narsingh, Graph Theory with Application to Engineering and Computer Science, Prentice Hall, India, 1979. • Vasishtha A. R. and Vasishtha A. K., Matrices, Krishna Prakashan, 2022. **Reference Books** • Grimaldi Ralph P. and Ramana B. V., Discrete and Combinatorial Mathematics: An Applied Introduction, Fifth Edition, Pearson Education, 2007. • Rosen Kenneth H. and Krithivasan Kamala, Discrete Mathematics and its Applications, McGraw Hill, India, 2019. • West Douglas B., Introduction to Graph Theory, Second Edition, Pearson Education, 2015 **Web Resources** • https://nptel.ac.in/courses/106103205 • https://nptel.ac.in/courses/111101115	

BCA-CC-112 Computer Architecture

Total Hours: 60 **Credits: 4**

Course objectives	• Understand the fundamental concepts of digital electronics and binary number systems, essential for computer architecture. • Learn the design and implementation of combinational circuits such as adders, multiplexers, and decoders, which form the basis of digital logic. • Explore the organization and design of basic computers, including instruction cycles, memory-reference instructions, and CPU architecture. • Understand the concept of parallel processing, pipelining, and memory organization for efficient computer processing and data management.	
Course outcomes	• Apply digital principles and number systems to solve problems related to digital logic and data representation. • Design and implement combinational and sequential circuits like adders, subtractors, flip-flops, and binary counters for practical applications. • Analyse the basic organization and design of computers, focusing on CPU architecture, instruction formats, and addressing modes. • Understand and implement parallel processing, pipelining techniques, and memory organization, improving computational efficiency.	
Unit	**Content**	**Hours**
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, 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, Excess3 Code, The Gray Code	**7**
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	**8**
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.	**7**

Unit IV	**Pipeline and Vector Processing:** • Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction 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.	**8**
Study Resources	**Text Books:** • Donald P Leach, Albert Paul Malvino, Goutam Saha- "Digital Principles & Applications" , Tata McGraw Hill Education Private Limited,2011Edition. • M. Morris Mano- "Computer System Architecture", Pearson/Phi, Third Edition. **Reference Books:** • William Stallings- "Computer Organization and Architecture", Pearson/PHI, Sixth Edition, • Andrew S. Tanenbaum- "Structured Computer Organization", PHI /Pearson 4th Edition, • M.V .Subramanyam, "Switching Theory and Logic Design", Laxmi Publications (P) Ltd. • Ikvinderpal Singh, Computer Organization Architecture, Khanna Book Publishing	

BCA-CC-113 Practical on Computer Architecture

Total Hours: 60 **Credits: 4**

Course objectives	• Understand the internal layout and components of a computer system, including SMPS, motherboard, RAM, HDD, and various ports. • Develop skills in identifying hardware specifications and troubleshooting common hardware problems related to RAM, SMPS, and the motherboard. • Learn how to configure BIOS settings and install additional hardware components like RAM. • Acquire hands-on experience in assembling a PC and installing software, such as operating systems and printers.
Course outcomes	• Identify and understand the layout of major hardware components within a computer system, such as the motherboard and various ports. • Diagnose and troubleshoot hardware issues, particularly with RAM, SMPS, and the motherboard, and identify the hardware specifications of a computer. • Demonstrate the ability to configure BIOS settings for USB and LAN functionality and expand system memory by adding additional RAM. • Install and configure operating systems and peripheral devices, including setting up a printer and resolving related issues.

Content

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. Verification of Truth Tables of J-K Flip-Flop using NAND/NOR gate
6. Familiarize the computer system layout: marking positions of SMPS, motherboard, FDD, HDD, CD, DVD and add on cards.
7. Identify the Computer Name and Hardware Specification (RAM capacity, Processor type, HDD, 32 bit/ 64 bit).
8. Identify and troubleshoot the problems of RAM, SMPS and motherboard.
9. Configure BIOS settings- disable and enable USB and LAN
10. Adding additional RAM to the system. (Expanding RAM size).
11. To Study mother board layout of a system.
12. Demonstrate the assembly of a PC.
13. Demonstration of various ports: CPU, VGA port, PS/2 (keyboard, mouse), USB, LAN, Speaker, Audio.
14. Install and configure windows OS.
15. To study the installation of Printer and trouble shooting

BCA-SEC-111 Problem Solving Techniques

Total Hours: 30 **Credits: 2**

Course objectives	• Understand the basics of problem-solving and algorithm development, focusing on generalization, analysis, and the role of data structures. • Learn structured programming concepts including selection, repetition, pseudocode, and flowcharts, and their translation into C language. • Develop skills to solve numerical problems like palindrome, prime numbers, and base conversions using C programming constructs. • Explore modular programming and array manipulation, focusing on recursion, searching, sorting, and matrix operations in C language.	
Course outcomes	• Apply problem-solving techniques such as breaking down problems into subproblems and understanding algorithm efficiency and correctness. • Implement structured programming concepts using C language constructs such as loops, conditionals, and good coding practices. • Solve mathematical and numerical problems using C programming techniques, including the use of operators, switch cases, and loops. • Implement modular programming and perform array operations, including searching, sorting, and matrix manipulation in C language.	
Unit	**Content**	**Hours**
Unit I	**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.	**7**
Unit II	**Structured Programming Concepts:** Sequence (Input/Output/Assignment), Selection (If, If-Else) And Repetition (For, While, Do-While) Statements, Control Structure Stacking and Nesting o Different Kinds of Repetitions: Entry Controlled, Exit Controlled, Counter Controlled, Definite, Indefinite and Sentinel-Controlled Repetitions. Pseudocode and Flowcharts. o 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. o 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. o C Language: Introduction To Programming Languages, Different Generations of Programming Languages. Typed Vs Typeless Programming Languages, History of C Language, An Empty C Program. o 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.	**8**

	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	**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	**7**
Unit IV	**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.	**8**
Study Resources	**Text Books** • Venkatesh, Nagaraju Y, Practical C Programming for Problem Solving, Khanna Book Publishing Company, 2024. • AICTE's Programming for Problem Solving (with Lab Manual), Khanna Book Publishing Company, 2024. • 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** • Brian W. Kernighan and Dennis Ritchie, The C Programming Language, 2nd edition, Pearson, 2015. • Jeri Hanly and Elliot Koffman, Problem Solving and Program Design in C, 8th edition, Pearson, 2015.	

BCA-SEC-112Problem Solving Techniques: Lab Problems

Total Hours: 30 **Credits: 2**

Course objectives	• Understand basic terminology of computers, problem solving, programming Languages and their evolution (Understand) • Create specification from problem requirements by asking questions to disambiguate the requirement statement. (Create) • 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) • Translate an algorithm into a C computer program (Create) • Testing and analysing programs using debugging tools. (Analyze)	
Course outcomes	• Learn algorithmic thinking to design efficient solutions for computational challenges. • Understand and apply structured problem-solving approaches for complex tasks. • Develop critical thinking and logical reasoning skills to approach various problems.	
Unit	**Content**	**Hours**
	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 the range [1,n]? 5. Display first mathematical tables, each table up to 10 rows? 6. Display the first n (n > 0) terms of the fibonacci sequence? 7. Extract digits of an integer number (left to right and right to left)? 8. Check if a given positive integer number is a palindrome or not? 9. 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? 10. Check if a given positive integer number is a prime number or not? 11. Compute prime factors of a positive integer number? 12. Check if two positive integer numbers are amicable numbers or not? 13. Check if a given positive integer number is a perfect number or not? 14. Check if a given positive integer number Armstrong number or not? 15. Design a modularized algorithm/program to check if a given positive integer number is a circular prime or not? 16. Design a modular algorithm/program which reads an array of n integer elements and outputs median? 17. Implement your own string length and string reversal functions? 18. Design algorithm/program to perform matrix operations addition, subtraction and transpose? 19. Write a recursive program to count the number of digits of a positive integer number? 20. To print a sequence of numbers entered using sentinel controlled repetition in reverse order?	

Study Resources	**Text Books** • Venkatesh, Nagaraju Y, Practical C Programming for Problem Solving, Khanna Book Publishing Company, 2024. • AICTE's Programming for Problem Solving (with Lab Manual), Khanna Book Publishing Company, 2024. • 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** • Brian W. Kernighan and Dennis Ritchie, The C Programming Language, 2nd edition, Pearson, 2015. • Jeri Hanly and Elliot Koffman, Problem Solving and Program Design in C, 8th edition, Pearson, 2015.	

BCA-AEC-111 Business Communication – I

Total Hours: 60 **Credits: 4**

Course objectives	• **Develop Communication Skills**: Understand communication concepts, processes, technological impacts, and workplace communication channels. • **Master Business Correspondence**: Apply effective principles for business letters, emails, and memos in various formats and types. • **Enhance Writing Skills**: Improve report and proposal writing, summarization, and document proofreading and editing. • **Strengthen Public Relations and Presentation**: Learn public relations functions, and improve writing, comprehension, speaking, and crisis management skills.	
Course outcomes	• **Effective Communication**: Students will be able to communicate effectively using various methods and channels in professional settings. • **Proficient Business Correspondence**: Students will demonstrate proficiency in drafting clear and professional business letters, emails, and memos. • **Advanced Writing Skills**: Students will produce well-organized reports and proposals, and exhibit strong summarization and editing abilities. • **Improved Public Relations and Presentation**: Students will exhibit competence in public relations practices, and deliver effective written and oral presentations, including managing crises.	
Unit	**Topic Particular**	**Hours**
Unit I	**Unit 1: Theory of Communication** • **Concept of Communication –** Meaning, Definition, Process, Need, Feedback, Emergence of Communication as a key concept in the Corporate and Global world • **Impact of technological advancements on Communication**- Types- Internet, Blogs, E-mails, Moodle, Social media (Face book, X (formerly Twitter) &WhatsApp)Advantages and Disadvantages • **Communication at work place**– Channels-- Formal and Informal— Vertical, Horizontal, Diagonal, Grapevine. Methods: Verbal and Nonverbal. Characteristics of Non-verbal Communication • **Problems in Communication /Barriers to Communication --** Physical/ Semantic/Language / Socio-Cultural / Psychological / Barriers Ways to Overcome these Barriers • **Listening Skills –** Importance of Listening Skills, Obstacles to listening, cultivating good Listening Skills	**15**
Unit II	**Unit 2: Business Correspondence** • **Theory of Business Letter Writing --** Parts, Structure, Layouts— Full Block, Modified Block, Semi - Block • Principles of Effective Letter Writing: Types Inquiry Letter, Complaint Letter, Order Letter, Cover Letter, Thanking Letter, Order Letter, Recommendation Letter • **Principles of effective Email Writing:** Significance, Structure, Etiquette • **Memos**- Meaning, Purpose, Structure • **Personnel Correspondence –** Statement of Purpose, Job Application	15

	Letter and Resume. Letter of Acceptance of Job Offer, Letter of Resignation ● **Digital Signature:** Maintaining Records and Documents [Letter of Appointment, Promotion and Termination, Letter of Recommendation]	
Unit III	**Unit 3: Language and Writing Skills** ● Reports and Business Proposals – Objective, Purpose, Logical Flow and Organisational, Supporting Data and Evidence ● Parts, Types-Informational, Analytical, Research, Progress ● Feasibility Reports, Investigative Reports, Drafting Business Proposals ● Summarization– Identification of main and supporting/sub points and presenting these in a cohesive manner ● Proof Reading and Editing	**15**
Unit IV	**Public Relation** ● Meaning, Function, Internal and External Measures ● Paragraph Writing ● Reading Comprehension ● Listening Comprehension ● Crises Management ● Speaking Skills: Presenting News items, Dialogue & Speech delivery.	**15**
Study Resources	● Chaturvedi, P. D., & Chaturvedi, M. (2023). Business Communication: Skills, Concepts, and Applications. Pearson. ● Raman, M., & Sharma, S. (2023). Technical Communication: Principles and Practice. Oxford University Press. ● Singh, A. (2023). Business Communication: Connecting in a Digital World. Cengage Learning. ● Ghosh, R. (2023). Business Communication: Process and Product. McGraw Hill. ● Kaul, A. (2023). Effective Business Communication. PHI Learning. ● Rai, U., & Rai, S. M. (2022). Business Communication. Himalaya Publishing House. ● Bedi, R., & Aruna, K. (2022). Business Communication. Vrinda Publications.	

SEMESTER –II

BCA-CC-121 Mathematics Foundation to Computer Science – II

Total Hours: 30 **Credits: 2**

Course objectives	• To introduce basic logical operations, truth tables, and methods of proofs, enabling the application of logical reasoning and formal methods in problem-solving. • To familiarize with foundational algebraic structures such as semi-groups, monoids, and groups, focusing on their properties and applications. • To provide knowledge of numerical methods for solving algebraic and transcendental equations, as well as performing interpolation and integration. • To introduce optimization techniques, including linear programming and transportation problems, facilitating the formulation and solving of real-world problems.	
Course outcomes	• Demonstrate the ability to construct truth tables, identify logical equivalences, and apply methods of proofs such as modus ponens, modus tollens, and mathematical induction. • Define and illustrate the properties of algebraic structures like groups, subgroups, and cyclic groups, and apply these concepts to solve mathematical problems. • Apply methods like bisection and Newton-Raphson to solve algebraic and transcendental equations, along with using numerical interpolation and integration techniques. • Formulate and solve linear programming problems using graphical and simplex methods, and determine feasible and optimal solutions for transportation problems using various methods.	
Unit	**Content**	**Hours**
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.	**08**
Unit II	**Algebraic Structures:** • Semi-group, Monoid, Group, Subgroup, Cyclic group.	**07**
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.	**07**
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	**08**

	solution, MODI method for finding optimum solution.	
Study Resources	**Text Books** • Kolman B., Busby R. and Ross S., Discrete Mathematical Structures, 6th Edition, Pearson Education, 2015 • Sastry S. S., Introductory Methods of Numerical Analysis, Fifth Edition, PHL, 2022. Model curriculum for UG Degree in BCA 44 • Taha Hamdy A., Operations Research: An Introduction, Eighth Edition, Pearson Prentice Hall, 2003. • S.B. Singh, Discrete Structures, Khanna Book Publishing, 2023 (AICTE Recommended Textbook) **Reference Books** • Rosen Kenneth H. and Krithivasan Kamala, Discrete Mathematics and its Applications, McGraw Hill, India, 2019. • Chakravorty J. G. and Ghosh P. R., Linear Programming and Game Theory, Moulik Library, 2017. • Sharma J. K., Operations Research: Theory and Applications, Fourth Edition, Macmilllan Publishers, 2007. **Web Resources** • https://nptel.ac.in/courses/111107127 • https://www.math.iitb.ac.in/~siva/si50716/SI507lecturenotes.pdf	

BCA-CC-122 Data Structures

Total Hours: 60 **Credits: 4**

Course objectives	• Understand the fundamental concepts of Data Structures and their applications. • Develop problem-solving skills using Data Structures. • Implement Data Structures using C programming language.	
Course outcomes	• Analyze the efficiency of algorithms using different data structures. • Learn to apply data structures in algorithm design and real-world applications. • Develop problem-solving skills through effective use of data structures in various scenarios.	
Unit	**Content**	**Hours**
Unit I	**Introduction and Overview:** • Definition, Classification and Operations of Data Structures. Algorithms: Complexity, Time-Space Trade-off. **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.	**15**
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.	**15**
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.	**15**

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.	**15**
Study Resources	**Text Books** • R.B. Patel, "Expert Data Structures with C", Khanna Book Publishing Company, 2023 (AICTE Recommended Textbook) • Seymour Lipschutz, "Data Structures with C", Schaum's Outlines, Tata McGraw-Hill, 2011. • Yashavant Kanetkar, "Data Structures Through C", 4th Edition, BPB Publications, 2022. **Reference Books** • Reema Thareja, "Data Structures Using C", Second Edition, Oxford University Press, 2014. • Ellis Horowitz, Sartaj Sahni, and Susan Anderson-Freed, "Fundamentals of Data Structures in C", Second Edition, Universities Press, 2007. **Web Resources** • GeeksforGeeks - Data Structures Tutorial • 2. Khan Academy - Algorithms Cours	

BCA-CC-123 Practical on Data Structures

Total Hours: 60 **Credits: 2**

Course objectives	• To develop the ability to perform various operations on arrays, such as insertion, deletion, searching, sorting, and merging, through hands-on programming tasks. • To gain practical skills in manipulating singly, doubly, and circular linked lists, performing operations such as insertion, deletion, and searching. • To enable the implementation of stack and queue operations using both arrays and linked lists, with a focus on problem-solving through data structures. • To apply mathematical operations such as matrix addition, subtraction, multiplication, and polynomial addition using linked lists in programming environments.
CourseOutcomes	• Demonstrate the ability to implement insertion, deletion, searching (linear and binary), sorting (bubble, selection, and insertion), and merging of arrays using appropriate algorithms. • Perform insertion, deletion, and search operations on singly, doubly, and circular linked lists, and apply linked lists in practical scenarios such as polynomial addition. • Implement stack operations using arrays and linked lists, and perform simple and circular queue operations, demonstrating proficiency in handling linear data structures. • Successfully implement programs to add, subtract, and multiply matrices, and perform polynomial addition using linked lists, applying mathematical operations in computational contexts.
Sr. No.	**Practical's**
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.
4	Write a program to sort an array using Selection Sort.
5	Write a program to sort an array using Insertion Sort.
6	Write a program to add, subtract and 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 implement stack operations using link list.

11	Write a program to evaluate a postfix expression using a stack.
12	Write a program to implement simple queue operations using an array.
13	Write a program to implement circular queue operations using an array.
14	Write a program to perform the following operations on a binary search tree. (a) Preorder Traversal (b) Inorder Traversal (c) Postorder Traversal
15	Write a program to perform insertion operation in a binary search tree.

BCA-CC-124 Operating Systems

Total Hours: 60 **Credits: 4**

Course objectives	• To introduce the evolution, types, structure, and components of operating systems, including batch, multiprogramming, and real-time systems. • To explain process management, including scheduling, multithreading, and memory management techniques like paging and segmentation. • To provide insight into process synchronization, inter-process communication, and deadlock handling. • To cover the principles of I/O hardware and disk scheduling, focusing on efficient resource management.	
Course outcomes	• Understand the architecture, services, and types of operating systems, and explain concepts like multiprogramming and time-sharing. • Implement scheduling algorithms and apply memory management techniques such as paging and virtual memory, understanding fragmentation. • Solve synchronization issues, use semaphores, and apply deadlock prevention and recovery strategies like the Banker's algorithm. • Apply disk scheduling algorithms and explain the role of hardware in managing I/O devices for better system performance.	
Unit	**Content**	**Hours**
Unit I	**Operating Systems Overview:** o Definition, Evaluation of O.S, 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:** o Operating system services and systems calls, system programs, operating system structure, operating systems generations.	**15**
Unit II	**Process Management:** o Process Definition, Process states, Process State transitions, Process Scheduling, Process Control Block, Threads, Concept of multithreads, Benefits of threads, Types of threads. **Process Scheduling:** o Definition, Scheduling objectives, Scheduling algorithms, CPU scheduling Pre-emptive and Non-pre-emptive Scheduling algorithms (FCFS, SJF and RR), Performance evaluation of the scheduling Algorithms	**15**
Unit III	**Process Synchronization:** o Introduction, Inter-process Communication, Race Conditions, Critical Section Problem, Mutual Exclusion, Semaphores, Monitors. **Deadlocks:** o System model, deadlock characterization, deadlock prevention, avoidance, Banker's algorithm, Deadlock detection, and recovery from deadlocks.	**15**
Unit IV	**Memory Management:** o Logical and Physical address map, Swapping, Memory allocation, MFT, MVT, Internal and External fragmentation and Compaction, Paging, Segmentation. **Virtual Memory:** o Demand paging, Page Replacement algorithms, Allocation of frames,	**15**

	thrashing. **I/O Management:** ○ Principles of I/O Hardware: Disk structure, Disk scheduling algorithms.	
Study Resources	**Text Books:** • Ekta Walia, Operating Systems Concepts, Khanna Publishing House, 2022 (AICTE Recommended Textbook) • Abraham Silberschatz, Peter Baer Galvin, Greg Gagne (2006), Operating System Principles, 7th edition OR Later edition, Wiley India Private Limited, New Delhi. • Stallings (2006), Operating Systems, Internals and Design Principles, 5th edition, Pearson Education, India. **Reference Books:** • Andrew S Tanenbaum, Modern Operating Systems, Third Edition, Prentice Hall India.Model curriculum for UG Degree in BCA 48 • Sumitabha Das, UNIX Concepts and Applications, 4th Edition, Tata McGraw-Hill.	

BBA-CC-125 Practical on Operating Systems

Total Hours: 60 **Credits: 2**

Course objectives	• To implement key CPU scheduling algorithms like FCFS, SJF, and Round Robin using C programming. • To simulate deadlock avoidance (Banker's algorithm) and solve synchronization problems (Producer-Consumer using semaphores). • To implement memory management techniques such as Paging, Segmentation, and contiguous memory allocation (Best Fit, First Fit). • To simulate inter-process communication using pipes and FIFOs, and implement page replacement techniques like FIFO.
CourseOutcomes	• Implement FCFS, SJF, and Round Robin algorithms and analyse performance (turnaround and waiting time). • Apply the Banker's algorithm to avoid deadlocks and solve the Producer-Consumer problem using semaphores. • Implement Paging, Segmentation, and memory allocation techniques like Best Fit and First Fit. • Demonstrate IPC using pipes and FIFOs, and implement FIFO page replacement.
Sr. No.	**Practical's**
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

BCA-SEC-121 Object Oriented Programming using Java

Total Hours: 30 **Credits: 2**

Course objectives	• To introduce the object oriented programming system concepts • To introduce syntax and semantics of Java programming language • To develop modular programs using Java • To setup JDK environment to create, debug and run Java programs	
Course outcomes	• Understand the principles of object-oriented programming (OOP) and its benefits. • Learn to design and implement programs using Java classes, objects, and inheritance. • Build real-world applications using Java, leveraging GUI development and advanced features.	
Unit	**Content**	**Hours**
Unit I	**Fundamentals of Object-Oriented Programming:** o Basic Concepts of Object-Oriented Programming (OOP), Benefits and Applications of OOP. **Java Evolution:** o Java Features, Difference between Java, C and C++, Java and Internet, Java Environment. **Overview of Java Language:** o Introduction to Simple Java Program, Use of Comments and Math function, Application of two classes, Program Structure, Java Tokens and statements, Implementing Java program and JVM, Command Line Arguments.	**08**
Unit II	**Constants, Variables and Data Types:** o Constants, Variables, Data Types, Declaration of Variables, Giving values to Variables, Symbolic Constants, Typecasting. **Operators & Expressions**: o 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:** o Decision Making with Control Statements, Looping statements, Jump in loops, Labelled loops.	**07**
Unit III	**Classes, Objects and Methods:** o Defining Class, Methods Declaration,Constructors, Methods Overloading, Overriding Methods, Inheritance. **Arrays, Strings and Vectors:** o 1D arrays, Creating an Array, 2D arrays, Strings,Vectors, Wrapper Classes, Enumerated Types. **Inheritance:** o Defining, extending classes, and Implementing Interfaces. Multipleinheritance and polymorphism.	**08**

Unit IV	**Packages:** ○ Basics of packages, System packages, Creating and accessingpackages, 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.	**07**
Study Resources	**Text Books** • Balaguruswamy E. (2023). Programming with JAVA: A Primer. 7th edition. India: McGraw Hill Education • Schildt, H. (2022). Java: The Complete Reference. 12th edition. McGraw-Hill Education. **Reference Books** • Arunesh Goyal, The Essentials of JAVA, Khanna Book Publishing Company Private Limited, 2012. • Tanweer Alam, Core JAVA, Khanna Book Publishing Company Private Limited, 2015. • Y. Daniel Liang, Introduction to Java Programming, 7th Edition, Pearson, 2008. • S. Malhotra and S. Choudhary, Programming in Java, 2nd Edition, Oxford University Press, 2014. **Web Resources** • https://www.w3schools.com/java/. • http://www.java2s.com/. • https://onlinecourses.nptel.ac.in/noc22_cs47/preview	

BCA-SEC-122 Practical on Object Oriented Programming using Java

Total Hours: 60 **Credits: 4**

Course objectives	• Learn fundamental Java concepts like input handling, command line arguments, and basic arithmetic. • Solve problems using recursion, array operations, and matrix manipulation. • Understand object-oriented principles like inheritance and exception handling. • Explore Java's exception handling, file handling, and package management for real-world tasks.
Course Outcomes	• Create programs that handle input/output, perform arithmetic, and use command line arguments. • Write programs for array operations, Fibonacci series, and factorial using recursion and non-recursion. • Apply inheritance, method overriding, and exception handling to solve problems like mathematical operations and packages. • Use exception handling for errors like divide-by-zero and input duplicates, and manage file operations to count characters, words, and lines.
Sr. No.	**Practical's**
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.
11	Write a Java program that displays the number of characters, lines and words in a text.
12	Write a Java package program for the class book and then import the data from the package and display the result.
13	Write a Java program for demonstrating the divide by zero exception handling.
14	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.
15	Write a Java program that reads a list of integers from the user and throws an exception if any numbers are duplicates.