



**Darrang College  
(Autonomous),  
Tezpur-784001**

**Syllabus for  
FYUGP  
BCA**

**Approved by :**

**Board of Studies meeting held on 29<sup>th</sup> July, 2025**

**&**

**Academic Council vide Resolution no. 04, dated- 12/08/2025**

## **PROGRAMME STRUCTURE (FYUGP BCA)**

Eligibility: 10+2 Qualified

| <b>Semester</b> | <b>Course Name</b>                             | <b>Credit</b> | <b>L+T+P</b> | <b>Course Type</b> |
|-----------------|------------------------------------------------|---------------|--------------|--------------------|
| <b>1</b>        | Introduction to C- Programming                 | 4             | <b>3+0+1</b> | <b>Core</b>        |
|                 | Mathematics 1                                  | 4             | <b>3+1+0</b> | <b>Core</b>        |
|                 | Computer Fundamentals and Application Software | 3             | <b>2+0+1</b> | <b>SEC</b>         |
|                 | Alternative English                            | 4             |              |                    |
|                 | MDC 1                                          | 3             |              |                    |
|                 | VAC                                            | 2             |              |                    |
| <b>2</b>        | Data Structures & Algorithms Using C           | 4             | <b>3+0+1</b> | <b>Core</b>        |
|                 | Digital Logic                                  | 4             | <b>3+1+0</b> | <b>Core</b>        |
|                 | Web Designing                                  | 3             | <b>2+0+1</b> | <b>SEC</b>         |
|                 | English Communication                          | 4             |              |                    |
|                 | MDC 2                                          | 3             |              |                    |
|                 | VAC 2                                          | 2             |              |                    |
| <b>3</b>        | Computer Organization and Architecture         | 4             | <b>4+0+0</b> | <b>Core</b>        |
|                 | Object Oriented Programming through C++        | 4             | <b>3+0+1</b> | <b>Core</b>        |
|                 | Automata Theory and Languages                  | 4             | <b>3+0+1</b> | <b>Core</b>        |
|                 | SEC (Not Decided)                              | 3             | <b>2+0+1</b> | <b>SEC</b>         |
|                 | MDC 3                                          | 3             |              |                    |
|                 | VAC 3                                          | 2             |              |                    |
| <b>4</b>        | Database Management System                     | 4             | <b>3+0+1</b> | <b>Core</b>        |
|                 | Mathematics 2                                  | 4             | <b>4+0+0</b> | <b>Core</b>        |
|                 | Software Engineering                           | 4             | <b>4+0+0</b> | <b>Core</b>        |
|                 | System Software                                | 4             | <b>3+0+1</b> | <b>Core</b>        |
|                 | Internship                                     | 4             |              | <b>Core</b>        |
| <b>5</b>        | Operating System                               | <b>6</b>      | <b>5+0+1</b> | <b>Core</b>        |
|                 | Artificial Intelligence                        | 4             | <b>3+0+1</b> | <b>Core</b>        |
|                 | Programming in Python                          | 4             | <b>3+0+1</b> | <b>Core</b>        |
|                 | Minor Project                                  | 4             |              | <b>Core</b>        |
| <b>6</b>        | Computer Networks                              | <b>6</b>      | <b>5+1+0</b> | <b>Core</b>        |
|                 |                                                | 4             | <b>3+0+1</b> | <b>Core</b>        |
|                 |                                                | 4             |              | <b>Core</b>        |
|                 | MAJOR PROJECT                                  | 8             |              | <b>Core</b>        |

### **Programme outcomes of FYUGP BCA Programme**

The completion of the BCA Programme shall enable a student to:

- i) To communicate technical information both orally and in writing
- ii) Apply the knowledge gained in core courses to a broad range of advanced topics in computer science, to learn and develop sophisticated technical products independently.
- iii) To design, implement, and evaluate computer-based system, process, component, or program to meet desired needs by critical understanding, analysis, and synthesis
- iv) Identify applications of Computer Science in other fields in the real world to enhance the career prospects
- v) Realize the requirement of lifelong learning through continued education and research.
- vi) Use the concepts of best practices and standards to develop user interactive and abstract application
- vii) Understand the professional, ethical, legal, security, social issues and responsibilities

## **Introduction to C-Programming (BCA-CR-01014)**

1. Learning Outcomes: At the end of the course, students will be able to:
  - (a) Understand the basics of C programming like data types and operators
  - (b) Understand and write program in C to implement conditions, loops, functions
  - (c) Work on arrays, strings and basic file operations
2. Prerequisites: NIL
3. Semester: 1
4. Course type: Compulsory
5. Theory credit: 3
6. Practical credit: 1
7. Number of required hours:
  - (a) Theory: 45 hours (45 classes)
  - (b) Practical: 30 hours (15 classes)

### **Detailed Syllabus (Theory)**

#### Unit 1: Basics of C Programming

(10 Lectures)

Introduction to programming languages. Comparative study of different types of Programming languages. Translator and types. Structure of a C program. Introduction to Header files. Main function and a simple program execution. Compiling and executing a program. C tokens – keywords, identifiers, constants, operators. Statements and expressions in C. Basic data types in C - integers, float, double, character, void. Size and range of data types. Variables. Storage Class. Constants – integer constant, real constant, character constant, string constant. Declaration and initialization of variables and constants. Assigning values to variables. Operators in C – binary and unary operators. Arithmetic, assignment, logical, comparison, bitwise and conditional operators. Precedence and Associativity of operators. Input and output statements – getchar(), getch(), getche(), gets(), putchar(), puts(), scanf(), printf(), format specifiers. Typecasting.

#### Unit 2: Control Structures in C

(9 Lectures)

Need and use of control structure. Conditional statements – if, else, switch case, nested if-else. Loops – while loop, for loop, do-while loop. Using loop for counting iterations. Using while loop for indefinite iterations. Nested loops. Use of Break and continue statements.

### Unit 3: Arrays and Strings

(8 Lectures)

Introduction to Arrays. Types of Arrays. Declaration and initialization of arrays. Processing/Accessing array elements. Multidimensional arrays. Introduction to Strings. Declaration and initialization of strings. String input and output in C.

### Unit 4: Pointers and Functions

(9 Lectures)

Introduction to Pointers. Pointer declaration and initialization. Pointers and addresses. Pointers and Arrays. Basic concept of dynamic memory allocation, malloc(), calloc(). Introduction to user defined functions. Function declaration and definition. Return types of function. Function arguments. Function calling – call by value vs call by reference. Passing an array as argument to a function. Recursive Function.

### Unit 5: Introduction to Structures and Unions

(4 Lectures)

Basic concept of Structures and Unions in C. Structure declaration and initialization. Union declaration and initialization. Difference between structures and unions.

### Unit 6: File Processing and Preprocessor Directives

(5 Lectures)

Basic concept of file handling. Comparative study of Different types of files and their uses. Opening and closing file. Reading contents from file, writing to files. Random access to files. File pointers. Error handling in file operations. Preprocessor directives in C - #define, #ifdef, #include, #ifndef, and #endif directives. Using preprocessor directives to define constants and macros.

### List of Practical

1. Write a program to take input of two numbers and print their sum, product, difference.
2. Write a program to find the smallest or greatest of three numbers given as input.
3. Write a program to print the sum and product of digits of an integer.
4. Write a program to check whether an input number is palindrome or not.
5. Write a program to take a number representing a month and print the name of the month using switch case.
6. Write a program that calculates the grade of a student based on their marks in a subject using nested if-else statements. Also print the range of marks for each grade using switch case.
7. Write a program to take a number as input and print all the even numbers up to that number using while and for loop.
8. Write a program to ask the user for an input to stop a loop or continue repeating after printing the iteration count using a do-while loop.
9. Write a program to find the maximum, minimum, sum and average of n numbers without using array.

10. Write a program that takes two integers as input and finds their greatest common divisor (GCD) and LCM
11. Write a program that calculates the sum of the first n terms of the Fibonacci sequence, where n is entered by the user, using a for-loop.
12. Write a program that takes an integer as input and checks if it is a prime number.
13. Write a program to create an array with inputs from the user and print the same.
14. Write a program to read an integer and display the binary/octal equivalent of the number.
15. Write a program to take a matrix from the user and print the transpose of the same.
16. Write a program to find the sum, product of 2 matrices.
17. Write a program to take a string of length more than 10 and find the number of vowels in the string. Also print the position of the vowels in the string.
18. Write a program using pointers to copy a string to another string variable without using library function.
19. Write a program to check whether an input string is palindrome or not.
20. Write a program that swaps two numbers using pointers.
21. Write a program to calculate Factorial of a number (i) using recursion, (ii) using iteration
22. Write a program to find sum of n elements entered by the user. To write this program, allocate memory dynamically using malloc() / calloc() functions or new operator.
23. Write a function to accept two arrays as argument and returns their sum as an array.
24. Write a program to implement struct in C. Create a structure of Student with RNo, Name and other credentials with proper datatype and print the same.
25. Write a program to implement union in C. Create a structure of Person with Pid, Name and other credentials with proper datatype and print the same.
26. Write a C program that opens a file for reading and displays the contents of the file in binary mode and text mode.
27. Write a C program that opens a file for reading and displays the contents of the file line by line on the screen.
28. Write a C program that opens a file in append mode and allows the user to add text to the end of the file.

## **Mathematics 1 (BCA-CR-01024)**

1. Learning Outcomes: After successful completion of this course, students will be able to:
  - (a) Learn the concepts of set, relation, and function from Computer Science point of view.
  - (b) Know how to view a table/database as an n-ary relation.
  - (c) Learn what a matrix is and relate it with arrays used in programming.
  - (d) Understand determinants and how determinants are used in solving simultaneous equations.
  - (e) Get familiar with statistical and probabilistic measures that are used in computation related software/packages.
2. Prerequisites: NIL
3. Semester: 1
4. Course type: Compulsory
5. Theory credit: 4
6. Number of required hours: Theory: 60 hours (60 classes)

### **Detailed Syllabus (Theory)**

#### **UNIT 1: Sets, Relations and Functions (13 Lectures)**

Sets: definition of set, Representation of set, Different types of set, cardinality of sets, finite, countable and infinite sets. Properties of Set Operations on sets, Venn diagram. Relations: Definition and properties of binary relations, closures of relations, equivalence relations. Functions: Definition of function, one-to-one and onto, principles of mathematical induction.

#### **UNIT 2: Matrices (17 Lectures)**

Definition and different types of matrices, row and column operations; vectors and matrices, Addition, subtraction and multiplication of matrices, Properties of matrix operations, Existence of additive and multiplicative identity and additive inverse of a matrix. Representing relations using matrices. Transpose of a matrix and its properties. Symmetric and skew symmetric matrices, Elementary transformation of a matrix, Invertible matrices.

#### **UNIT 3: Determinants (17 Lectures)**

Determinant of a square matrix, minor, cofactor, Adjoint of a matrix and matrix inversion. Inverse of a matrix using elementary transformation. Rank of a matrix and determination of rank of a matrix. Eigen values and Eigen vectors of a matrix. Cayley-Hamilton theorem

– Cramer's rule, Consistency of a system of linear non-homogenous equations and existence of solutions, Solutions of simultaneous linear equations by Gaussian elimination method, Gauss Jordan Method.

#### UNIT 4: Statistics and Probability

(13 Lectures)

Data, Attributes, and variables; Frequency distribution, Cumulative frequency. Graphical representation of Frequency distribution: Histogram, Frequency Polygon, Frequency Curve and Cumulative Frequency curves (Ogive). Bar Diagram, Subdivided Bar Diagram, Pie diagrams. Measures of central tendency-Mean, Median and Mode. Measures of variation – Range, Interquartile range, Standard Deviation and Variance.

Sample space, events, random variables, basic probability. Conditional Probability and Bayes theorem.

## **Computer Fundamentals and Application Software (BCA-SEC-01013)**

1. Learning Outcomes: At the end of the course, students will be able to:
  - (a) Understand about the different hardware and software components of the computer and their functions.
  - (b) Use different Application Software for day-to-day activities.
2. Prerequisites: NIL
3. Semester: 1
4. Course type: Compulsory
5. Theory credit: 2
6. Practical credit: 1
7. Number of required hours:
  - (a) Theory: 30 hours (30 classes)
  - (b) Practical: 30 hours (15 classes)

### **Detailed Syllabus (Theory)**

#### **UNIT 1: Introduction to Computers and Number Systems (10 Lectures)**

Number system, decimal, binary, octal and hexadecimal number system, conversion among number systems, definition of computer, basic components of computer, bus, evolution of computers, Generations of computers, classification of computers, data representation in a computer, ASCII, Unicode.

#### **UNIT 2: Memory and Storage Devices (7 Lectures)**

Memory, memory hierarchy, registers, general purpose and special purpose registers, primary and secondary memory, volatile and non-volatile memory, semiconductor memory, SRAM and DRAM, Read Only Memory, magnetic storage devices, optical storage devices, solidstate drive, flash memory.

#### **UNIT 3: Input and Output Devices (5 Lectures)**

Input device, keyboard, keyboard layouts, pointing devices, mechanical and optical mouse, scanner, hand-held and flat-bed scanners, OMR, OCR, MICR, digital camera, touch pad, trackball, joystick, digitizer, digital microphone. Monitor, LCD, LED, plasma monitor, printers, impact printers, non-impact printers, dotmatrix printers, inkjet printers, laser printers, thermal printers, plotters, voice output systems, projector,

## Unit 4: Languages and Software

(8 Lectures)

Software; Types of software. System Software and different types. Operating System, Device driver. Application Software; Free and Open-Source Software. Comparative study between Open-Source Software and Proprietary Software. Language Translator. Difference between Compiler and Interpreter.

### **Practical Components:**

- |                                         |                      |
|-----------------------------------------|----------------------|
| A. Introduction to Libre Office Writer  | (12 Practical Hours) |
| B. Introduction to Libre Office Calc    | (12 Practical Hours) |
| C. Introduction to Libre Office Impress | (06 Practical Hours) |

## **Data Structure & Algorithm Using C (BCA-CR-02014)**

1. Learning Outcomes: At the end of the course, students will be able to:

- a) Understand and apply the fundamental data structures and algorithms – such as arrays, linked lists, stacks, queues, trees, sorting and searching algorithms using C programming language.
- b) Analyze the time and space complexity of different algorithms and choose the appropriate algorithm for a given problem.
- c) Develop efficient algorithms to solve various computational problems by utilizing data structures and algorithms covered in the course.

2. Prerequisites: Basic Knowledge of C programming

3. Semester: 2

4. Course Type: Compulsory

5. Theory Credit: 3

6. Practical Credit: 1

7. No of required hours:

- a) Theory: 45 hours (45 Classes)
- b) Practical: 30 hours (15 Classes)

### **Detailed Syllabus (Theory)**

#### **UNIT 1: Data Structures Overview**

(8 Lectures)

Concepts of Data Types, Abstract Data Type, Data Structure, Fundamental and Derived Data Types. Importance of data structures. Array as a data structure (characteristics, advantages, disadvantages). Representation of arrays – single and multidimensional. Address calculation of array element using column and row major ordering. Address translation functions for one & two dimensional arrays. Insertion and deletion in arrays. Use of arrays for large number representation.

#### **UNIT 2: Linked Lists**

(9 Lectures)

Initialization and implementation of structures. Structure and pointers. Self-referential structure. Introduction to linked lists. Singly linked list, doubly linked list, circular linked list. Operations on lists – creation, insertion, deletion, traversal.

#### **UNIT 3: Stacks and Queues**

(10 Lectures)

Definition of Stack and Queue. Representation of stacks and queues using arrays and linked lists. Stack operations – push, pop. Queue operation – enqueue, dequeue. Circular Queue, Priority Queue, Conversion of infix arithmetic expression containing arithmetic operators and parenthesis to postfix and prefix expression. Evaluation of postfix expression.

## UNIT 4: Tree

(8 Lectures)

Definition of Trees – General tree and Binary tree. Basic terminologies – parent, child, height, depth, leaf, node, internal nodes, external nodes. Brief concept of Forest, ordered trees, strictly binary tree, complete binary tree. Representation of trees using arrays and linked lists. Binary tree traversal methods – pre-order, in-order, post-order. Recursive and non-recursive algorithms for traversal methods. Binary search trees. Operation on BST – creation, insertion and deletion of a node. Definition and characteristics of threaded binary trees.

## UNIT 5: Searching and Sorting

(5 Lectures)

Linear and binary search. Indexed search. Hashing. Sorting algorithms – Insertion Sort, Selection Sort, Bubble Sort, Merge Sort, Quick Sort.

## UNIT 6: Analysis of Algorithm and Complexity

(5 Lectures)

Complexity measures of an algorithm – Time and space complexity. Average case and worst case analysis. Asymptotic notation as a measure of algorithm complexity,  $O$  and  $\theta$  notations. Best Case, Worst Case and Average Case Analysis of sorting algorithms-Selection sort, Bubble sort, Insertion sort, Heap sort, Quick sort and Searching algorithms – linear search and binary search.

### List of Practical

1. Write a program to declare an array and initialize the values according to the user. Now ask the user for a number  $n$  and return the  $n$ th element from the array.
2. Implement linked list in a program by writing functions for the following:
  - a. Create a singly linked list of  $n$  nodes
  - b. Count the number of nodes in the list
  - c. Print the values of all the nodes
  - d. Add a node at first, last and  $k$ th position in the linked list
  - e. Delete a node from first, last and  $k$ th position
  - f. Search for an element in the list. If found, return the position of the node. If not found, return a negative value.
3. Write a program to implement doubly linked list.
4. Write a function to concatenate two linked lists.
5. Write a program to take a number  $k$  and split the linked list after  $k$ th position.
6. Write a program to merge two sorted linked lists.
7. Write a program to implement list of lists.
8. Write a program to implement stack using array. Use push and pop operations on the array representation of the stack. Check whether the stack is full or empty.

9. Write a program to implement stack using linked list. Use push and pop operations on the stack by inserting nodes and deleting nodes from the linked list. Also check if the stack is full or empty.
10. Write a program to evaluate a simple postfix expression using stack.
11. Write a program to convert a decimal number into binary number using stack.
12. Write a program to implement queue using array. Add new elements to the queue and remove elements from the queue represented by array. Check whether the queue is full or empty.
13. Write a program to implement queue using linked list. Add new elements to the queue and remove elements from the queue represented by linked list. Also check whether the queue is full or empty.
14. Implement binary search and linear search algorithms on arrays.
15. Implement following sorting algorithms: Bubble sort, Insertion sort, Selection sort.

## **Digital Logic (BCA-CR-02024)**

1. Learning Outcomes: After completing this course, students will have grasp of
  - a) Fundamental concepts of digital logic that will make their base to understand the concepts of computer architecture and organization.
2. Prerequisites: NIL
3. Semester: 2
4. Course type: Compulsory
5. Theory credit: 4
6. Number of required hours: Theory: 60 Hours (60 classes)

### **Detailed Syllabus (Theory)**

#### UNIT 1: Introduction to Number System (10 Lectures)

Binary numbers, octal and hexadecimal numbers. Conversion of number. 1's complement and 2's complement, representation of signed binary number: 1's complement, 2's complement and signed magnitude, subtraction with complements, arithmetic addition and subtraction of signed binary numbers, binary codes: BCD, Excess-3, error detection code: parity bit, error correction code: Hamming code, gray code, ASCII, EBCDIC.

#### UNIT 2: Boolean Algebra and Logic Gates (25 Lectures)

Definition of Boolean algebra, two valued Boolean algebra, duality principle, theorems and postulates of boolean algebra, precedence of boolean operators, boolean expression and Venn diagram, Boolean functions and truth tables, complement of a boolean function, min terms and max terms, canonical forms of a Boolean function, sum of min terms and its short notation, product of max terms and its short notation, conversion between canonical forms, standard form of a boolean function. Karnaugh Map (upto 4 variable), don't-care conditions. Logic Gates. NAND and NOR implementation of boolean functions.

#### UNIT 3: Combinational Circuits (12 Lectures)

Definition of combinational circuit, half adder, full adder, half subtractor, full subtractor, BCD-to-Excess-3 code converter, encoders and decoders, multiplexers, de-multiplexers.

## UNIT 4: Sequential Circuits

(13 Lectures)

Flipflops, RS flip flop, D flip flop, JK flip flop, T flip flop, master slave flip flop. State table of a sequential circuit, state diagram, characteristic tables of flip flops, Analysis of Clocked Sequential circuits, State Reduction and Assignment, Flip –Flop Excitation tables, Design procedure of clocked sequential circuit.

## **Web Designing (BCA-SEC-02013)**

1. Learning Outcome: At the end of the course, students will be able to:
  - (a) Understand the basic concept of designing web applications.
  - (b) Design basic well-structured web page using HTML and CSS
  - (c) Develop the ability to implement interactive elements and dynamic content using basic JavaScript.
2. Prerequisites: NIL
3. Semester: 2
4. Course type: Compulsory
5. Theory Credit: 2
6. Practical Credit: 1
7. Number of required hours:  
Theory: 30 Hours (30 classes)  
Practical: 30 Hours (15 classes)

### **Detailed Syllabus (Theory)**

Unit 1: Overview of the World Wide Web and the internet (4 Lectures)

A brief history of TCP/IP and the Internet, Internet services-email, telnet, ftp, Internet components, the birth of web, web page, home page, web site, Web browsers. Introduction to web servers and their architecture.

Unit 2: HTML (8 Lectures)

Basics of HTML, HTML tags and attributes, creating list in HTML, hyperlinks, multimedia, HTML forms, tables in HTML, frames in HTML, image maps, style sheets in HTML. DHTML, XML. Creating simple web pages using MS Word or other word processor. Conversion of document formats. Cascading Style sheets.

Unit 3: JavaScript (10 Lectures)

Basics of JavaScript. Creating interactive web pages using JavaScript.

Unit 4: Basics of php (8 Lectures)

Database connectivity through php. Creating web pages using php; storing, retrieving data to/from database