BENGALURU NORTH UNIVERSITY

TAMAKA, KOLAR- 563103

Curriculum/Syllabus
for
Undergraduate Programme

Bachelor of Science

With three Major Subject combination

Choice Based Credit System
As per State Education Policy - Karnataka

Major Subject: **COMPUTER SCIENCE**

Faculty of Science

(With Effect from Academic Year 2024-25)

BENGALURU NORTH UNIVERSITY
KOLAR - 563103

State Education Policy - 2024
(Semester Scheme)

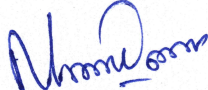### Curriculum Structure for Bachelor of Science in Computer Science - B.Sc. (CS)

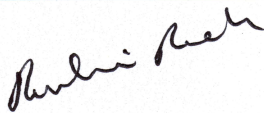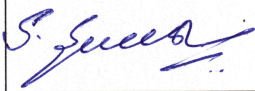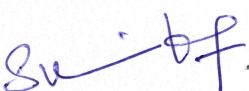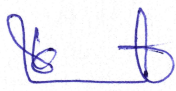## Syllabus for 1st and 2nd Semesters

With Effect From

Academic Year 2024 - 2025 and onwards

Curriculum Design/Syllabus Framing Committee

The following BOS members were present:

Sl.No.	Name and Address	Designation	Signature
1.	Dr. Murugan K, Associate Professor Department of Computer Science Government First Grade College K R Puram, Bengaluru-560036.	Chairperson	
2	Smt. Rashmi Rao K, Associate Professor Department of Computer Science Government First Grade College Hoskote, Bengaluru (Rural)	Member	
3	Dr. Sumanth S, Associate Professor Department of Computer Science Government College for Women Kolar – 563101	Member	
4	Dr. Rajendirakumar S, Associate Professor Department of Computer Science Government College for Women Kolar – 563101	Member	
5	Dr. Hamela K, Associate Professor Department of Computer Science Government First Grade College Malur, Kolar (Dist)	Member	
6	Mr. Manikandan S, Assistant Professor Department of Computer Science Government First Grade College K. R. Puram, Bengaluru	Member	
7	Dr. K. S. Manjunatha, Professor Department of Computer Science, Maharanis Science College for Women JLB Road, Mysore.	Member (External)	

Minutes:

1. The BoS members have approved the course structure and recommended (Theory and Lab).

2. The BoS members unanimously approved the proposed course structure for the newly introduced subjects.

3. The BoS members are accepted the detailed syllabi for the 1st and 2nd semester.

Regulations, Scheme of study and Examination for Bachelor of Science Degree in Computer Science Course Under Choice Based Credit System - Semester System (SEP Scheme (with effect from 2024 -2025)

R1**.**a) Title of the course: **Bachelor of Science in Computer Science.**

b) Duration of the Course: Durations of the undergraduate programme shall extend SIX semesters (Three academic years) for the regular Bachelor Degree.

c) Scheme of study: **i)** There shall be six theory papers and three practical from first semester to six semester.

d) Medium of Instruction: The medium of instruction shall be English.

e) Scheme of Examination: At the end of each semester there be University Examination of three hours duration in each of the theory paper/practical.

R2. Each semester shall be of 90 working days from the date of commencement of the each Semester.

R3. Attendance: As per Bengaluru North University regulations in force for science degree courses.

R4. A Candidate is allowed to carry over all the previous uncleared (failed) theory papers/Practical to subsequent semesters as per Bengaluru North University regulations in force for science degree courses.

R5. The maximum period for completion of the course shall be as per Bengaluru North University regulations in force for science degree courses.

R6. Eligibility for admission:

a) A candidate who has passed the two years Pre-University Examination conducted by the Pre-University Education Board in Karnataka in science combination with a minimum of 35% of marks.

b) A candidate who has passed Three years Diploma in Engineering of Government of Karnataka or any other examination considered as equivalent thereto shall be eligible for admission with minimum of 35% of marks in aggregate in all the semester /years.

c) Any student who has passed PUC–II in Science combination for other than Karnataka securing with a minimum of 35% of marks.

R7. The total number of students to be admitted to the course shall be decided by the Bengaluru North University.

R8. Results: Results of candidate shall be declared and the classes awarded as per the procedure followed by the Bengaluru North University.

R9. Power to Remove Difficulties

a) If may difficulty arises in giving effect to the provisions of these regulations, the Vice-Chancellor may be order make such provisions not inconsistent with the Act, Statutes, Ordinances or other Regulations, as appears to be necessary to expedient to remove the difficulty.

b) Every order made under this shall be subject to rectification by the appropriate University Authorities.

R10. The question paper pattern for theory paper has three sections. (80 Marks)

1. Section A includes 12 questions, students has to attend 10 questions. Each carries 2 Marks (10 * 2 = 20)
2. Section B includes 8 questions (question may contain sub questions), students has to attend 6 questions. each carries 5 Marks (6 * 5 = 30)
3. Section C includes 4 questions (question may contain sub questions), students has to attend 3 questions. each carries 10 Marks (3 * 10 = 30)

R11. Internal Assessment Scheme for Theory (4 Credits).

Assessment Criteria	Marks
Two test (Each carries 5 marks)	10
Assignment	5
Seminar	5
Total	20

R12. Internal Assessment Scheme for Theory (2 Credits).

Assessment Criteria	Marks
Test	5
Assignment	5
Total	10

R13. Internal Assessment Scheme for Practical (2 Credits).

Assessment Criteria	Marks
Test	5
Record	5
Total	10

R14. Evaluation Scheme for Practical Examination (SEE).

Assessment Criteria	Marks
Write up two programs (one from Section-A and Section-B)	20
Execution and output	15
Viva Voice based on Lab Activities	5
Total	40

Course Content for Bachelor of Science in Computer Science
B.Sc (CS)

Curriculum Structure

Program: B.Sc.   **Subject: Computer Science**

Semester	Course Code	Title of the Paper	Credits (L+P)	No.of Hours / Per Weak	Marks		Total Credits
					SEE	IA	
I	CS1T	Computer Fundamentals and Programming in C	4+0	4	80	20	6
	CS1P	C Programming Lab	0+2	4	40	10	
II	CS2T	Data Structures Using C	4+0	4	80	20	6
	CS2P	Data Structures Lab	0+2	4	40	10	

Semester: I
CS1T: Computer Fundamentals and Programming in C

Course Code: CS1T	Course Title: Computer Fundamentals and Programming in C
Course Credits: 04	Teaching Hours/Week: 04
Total Contact Hours: 60	Exam Duration : 3 Hours
Internal Assessment : 20 Marks	Semester End Exam: 80 Marks

Course Outcomes (COs):

After completing this course satisfactorily, a student will be able to:

- Confidently operate Desktop Computers to carry out computational tasks
- Understand working of Hardware and Software.
- Understand programming languages, number systems and peripheral devices.
- Read, understand and trace the execution of programs written in C language
- Write the C code for a given problem
- Perform input and output operations using programs in C
- Write programs that perform operations on arrays, strings, structures and pointers.

Course Content	Hours
Unit-I	15
Fundamentals of Computers: Introduction to Computers - Computer Definition, Characteristics of Computers, Evolution and History of Computers, Types of Computers, Basic Organisation of a Digital Computer; Number Systems – different types, conversion from one number system to another; Computer Codes – BCD, Gray Code, ASCII and Unicode; Boolean Algebra – Boolean Operators with Truth Tables; Types of Software – System Software and Utility Software; Computer Languages - Machine Level, Assembly Level & High Level Languages, Translator Programs – Assembler, Interpreter and Compiler; Planning a Computer Program - Algorithm, Flowchart and Pseudo code with Examples.	

Unit-II	15
Overview of C Language: History of C, Character set, C tokens, Identifiers, Keywords, Data types, Variables, Constants, Symbolic Constants , Operators in C, Hierarchy of Operators, Expressions, Type Conversions and Library Functions. Managing Input and Output Operation: Formatted and Unformatted I/O Functions. Decision making, branching and looping: Decision Making Statements - if Statement, if–else statement, nested if statement, else–if ladder, switch statement,	
Unit-III	15
Looping - while, do-while, for loop, Nested loop, break, continue, and goto statements. Functions: Function Definition, prototyping, types of functions, passing arguments to functions, Nested Functions, Recursive functions. Arrays: Declaring and Initializing, One Dimensional Arrays, Two Dimensional Arrays, Multi Dimensional Arrays - Passing arrays to functions.	
Unit-IV	15
Strings: Declaring and Initializing strings, Operations on strings, Arrays of strings, passing strings to functions. Storage Classes - Automatic, External, Static and Register Variables. Structures-Declaring and Initializing, Nested structure, Array of Structure, Passing Structures to functions, Unions, typedef, enum, Pointers – Declarations, Pointer arithmetic, Pointers and functions, Call by value, Call by reference, Pointers and Arrays, Arrays of Pointers, Pointers and Structures.	

Reference Books:

1. C: The Complete Reference, By Herbert Schildt.
2. C Programming Language, By Brain W. Kernighan
3. Kernighan & Ritchie: The C Programming Language (PHI)
4.P. K. Sinha & Priti Sinha: Computer Fundamentals (BPB)
5. E. Balaguruswamy: Programming in ANSI C (TMH)

CS1P: C Programming Lab

Course Code: CS1P	Course Title: C Programming Lab
Course Credits: 02	Teaching Hours/Week: 04
Total Contact Hours: 60	Exam Duration : 3 Hours
Internal Assessment : 10 Marks	Semester End Exam: 40 Marks

Section-A

1. Program to read radius of a circle and to find area and circumference.
2. Program to read three numbers and find the biggest of three.
3 Program to generate n primes.
4. Program to read a number, find the sum of the digits, reverse the number and check it for palindrome.
5. Program to read numbers from keyboard continuously till the user presses 999 and to find the sum of only positive numbers.
6. Program to read percentage of marks and to display appropriate message (Demonstration of else-if ladder).
7. Program to find the roots of quadratic equation (demonstration of switch Case statement).
8. Program to read marks scored by n students and find the average of marks (Demonstration of single dimensional array).
9. Program to remove duplicate element in a single dimensional Array.
10. Program to perform addition and subtraction of matrices.

Section-B

1. Program to find the length of a string without using built in function.
2. Program to demonstrate string functions.
3. Program to demonstrate pointers in C.
4. Program to check a number for prime by defining isprime( ) function.
5. Program to read, display and add two m x n matrices using functions.
6. Program to read, display and multiply two m x n matrices using functions.
7. Program to read a string and to find the number of alphabets, digits, vowels, consonants, spaces and special characters.
8. Program to Reverse a String using Pointer.
9. Program to swap two numbers using pointers.
10. Program to demonstrate student structure to read & display records of n students.

Semester: II

CS2T: Data Structures using C

Course Code: CS2T	Course Title: Data Structures using C
Course Credits: 04	Teaching Hours/Week: 04
Total Contact Hours: 60	Exam Duration : 3 Hours
Internal Assessment : 20 Marks	Semester End Exam: 80 Marks

Course Outcomes (COs):

After completing this course satisfactorily, a student will be able to:

• Describe how arrays, records, linked structures, stacks, queues, trees, and graphs are represented in memory and used by algorithms.

• Describe common applications for arrays, records, linked structures, stacks, queues, trees, and graphs.

• Write programs that use arrays, records, linked structures, stacks, queues, trees, and graphs.

• Demonstrate different methods for traversing trees.

• Describe the concept of recursion, give examples of its use.

• Discuss the computational efficiency of the principal algorithms for sorting and searching.

Course Content	Hours
Unit-I	15
Introduction and Overview: Definition, Elementary data organization, Data Structures, data structures operations, Abstract data types, Complexity of algorithms, asymptotic notations for complexity of algorithms. Arrays: Definition, Linear arrays, arrays as ADT, Representation of Linear Arrays in Memory, Traversing Linear arrays, Inserting and deleting elements, Multidimensional arrays, Representation of multidimensional arrays and sparse matrices.	
Unit-II	15
Sorting: Bubble sort, Insertion sort, Selection sort, Searching: Linear Search, Binary search. Linked list: Definition, Representation of Singly linked list in memory, Traversing a Singly linked list, Searching a Singly linked list, Memory allocation, Garbage collection, Insertion into a singly linked list, Deletion from a singly linked list; Doubly linked list, Header linked list, Circular linked list.	

Unit-III	15

Stacks – Definition, Array representation of stacks, Linked representation of stacks, Stack as ADT, Arithmetic Expressions: Polish Notation, Application of Stacks, Recursion, Towers of Hanoi, Implementation of recursive procedures by stack. Queues – Definition, Array representation of queue, Linked list representation of queues Types of queue: Simple queue, Circular queue, Double ended queue, Priority queue, Operations on Queues, Applications of queues

Unit-IV	15

Tree – Definitions, Binary trees, Representing binary trees in memory, Traversal of binary tree; preorder, inorder and postorder traversal; Binary Search Trees, Searching, Inserting and Deleting in a Binary Search Tree. Graphs: Graph terminology, Sequential representation of Graphs: Adjacency matrix, Graph Traversals: Breadth First Search and Depth First Search.

References:

1. Ellis Horowitz and Sartaj Sahni: Fundamentals of Data Structures
2. Tanenbaum: Data structures using C (Pearson Education)
3. Kamathane: Introduction to Data structures (Pearson Education)
4. Y. Kanitkar: Data Structures Using C (BPB)

CS2P: Data Structures Lab

Course Code: CS2P	Course Title: Data Structures Lab
Course Credits: 02	Teaching Hours/Week: 04
Total Contact Hours: 60	Exam Duration : 3 Hours
Internal Assessment : 10 Marks	Semester End Exam: 40 Marks

Section-A

1. Program to find GCD using recursive function
2. Program to display Pascal Triangle using binomial function
3. Program to generate n Fibonacci numbers using recursive function.
4. Program to implement Towers of Hanoi.
5. Program to implement dynamic array, find smallest and largest element of the array.
6. Program to create two files to store even and odd numbers.
7. Program to create a file to store student records.
8. Program to read the names of cities and arrange them alphabetically.
9. Program to sort the given list using selection sort technique.
10. Program to sort the given list using bubble sort technique.

Section-B

1. Program to sort the given list using insertion sort technique.
2. Program to sort the given list using quick sort technique.
3. Program to sort the given list using merge sort technique.
4. Program to search an element using linear search technique.
5. Program to search an element using recursive binary search technique.
6. Program to implement Stack.
7. Program to convert an infix expression to postfix.
8. Program to implement simple queue.
9. Program to implement linear linked list.
10. Program to display traversal of a tree.