



ಬೆಂಗಳೂರು ಉತ್ತರ ವಿಶ್ವವಿದ್ಯಾಲಯ

ಟಮಕ, ಕೋಲಾರ – 563103

CHOICE BASED CREDIT SYSTEM

*(Semester Scheme with Multiple Entry and Exit Options for
Under Graduate Course)*

SYLLABUS AS PER NEP GUIDELINES

SUBJECT: COMPUTER SCIENCE

2021-22 onwards

BENGALURU NORTH UNIVERSITY
KOLAR - 563103

National Education Policy - 2020
(Semester Scheme)

Model Curriculum Structure for Bachelor of Science
(Basic and Honours) Programmes with Computer Science
as Major and Minor Courses.

Model Syllabus for I and II Semester
and
Open Elective Courses in Computer Science.

Revised w.e.f.

Academic Year 2021 - 2022 and onwards

Model Curriculum Structure

Program: BSc (Basic and Honors)

Subject: Computer Science

1. Computer Science as MAJOR with another Subject as MINOR (Table IIA of Model Curriculum)

Sem	Discipline Specific Core Courses (DSC)	Hour of Teaching/ Week		Discipline Specific Elective Courses (DSE)/ Vocational Courses (VC)	Hour of Teaching/ Week
		Theory	Lab		
1	DSC-1: Computer Fundamentals and Programming in C DSC-1Lab: C Programming Lab	4	4		
2	DSC-2: Data Structures using C DSC-2Lab: Data structures Lab	4	4		
3	DSC-3: Object Oriented Programming Concepts and Programming in JAVA DSC-3Lab: JAVA Lab	4	4		
4	DSC-4: Database Management Systems DSC-4Lab: DBMS Lab	4	4		
5	DSC-5: Programming in PYTHON DSC-6: Computer Networks DSC-5Lab: PYTHON Programming lab DSC-6Lab: Computer Networks Lab	3 3	4 4	VC-1: Any one from Vocational Courses, Group – 1*	3
6	DSC-7: Internet Technologies DSC-8: Operating System Concepts DSC-7Lab: JAVA Script, HTML, CSS Lab DSC-8Lab: C# Programming Lab	3 3	4 4	VC-2: Any one from Vocational Courses, Group – 2* Internship:	3 2
7	DSC-9: Computer Graphics and Visualization DSC-10: Design and Analysis of Algorithms DSC-11: Software Engineering DSC-9Lab: Computer Graphics and Visualization Lab DSC-10Lab: Algorithms Lab	3 3 3	4 4	DSE-1: Any one from Discipline Specific Elective Courses, Group – 1** DSE-2: Any one from Discipline Specific Elective Courses, Group – 2** Research Methodology:	3 3 3
8	DSC-12: Artificial Intelligence and Applications DSC-13: Computer Organization and Architecture DSC-14: Data Warehousing and Data Mining DSC-12Lab: AI Lab	3 3 3	4	DSE-3: Any one from Discipline Specific Elective Courses, Group – 3** Research Project:	3 6

2. Computer Science as MAJOR with another Subject also as MAJOR (Table IIIA of Model Curriculum)

Sem	Discipline Specific Core Courses (DSC)	Hour of Teaching/ Week		Discipline Specific Elective Courses (DSE)	Hour of Teaching/ Week
		Theory	Lab		
1	DSC-1: Computer Fundamentals and Programming in C DSC-1Lab: C Programming Lab	4	4		
2	DSC-2: Data Structures using C DSC-2Lab: Data structures Lab	4	4		
3	DSC-3: Object Oriented Programming Concepts and Programming in JAVA DSC-3Lab: JAVA Lab	4	4		
4	DSC-4: Database Management Systems DSC-4Lab: DBMS Lab	4	4		
5	DSC-5: Programming in PYTHON DSC-6: Computer Networks DSC-5Lab: PYTHON Programming lab DSC-6Lab: Computer Networks Lab	3 3	4 4		
6	DSC-7: Internet Technologies DSC-8: Operating System Concepts DSC-7Lab: JAVA Script, HTML, CSS Lab DSC-8Lab: C# Programming Lab	3 3	4 4		
7	DSC-9: Computer Graphics and Visualization DSC-10: Design and Analysis of Algorithms DSC-11: Software Engineering DSC-9Lab: Computer Graphics and Visualization Lab DSC-10Lab: Algorithms Lab	3 3 3	4 4	DSE-1: Any one from Discipline Specific Elective Courses, Group – 1** DSE-2: Any one from Discipline Specific Elective Courses, Group – 2** Research Methodology:	3 3 3
8	DSC-12: Artificial Intelligence and Applications DSC-13: Computer Organization and Architecture DSC-14: Data Warehousing and Data Mining DSC-12Lab: AI Lab	3 3 3	4	DSE-3: Any one from Discipline Specific Elective Courses, Group – 3** Research Project:	3 6

3. Computer Science as MINOR with another Subject as MAJOR (As per Table IIA of Model Curriculum)

Semester	Discipline Specific Core Courses (DSC)	Hour of Teaching/Week	
		Theory	Lab
1	DSC-1: Computer Fundamentals and Programming in C DSC-1Lab: C Programming Lab	4	4
2	DSC-2: Data Structures using C DSC-2Lab: Data structures Lab	4	4
3	DSC-3: Object Oriented Programming Concepts and Programming in JAVA DSC-3Lab: JAVA Lab	4	4
4	DSC-4: Database Management Systems DSC-4Lab: DBMS Lab	4	4
5	DSC-5: Programming in PYTHON DSC-5Lab: PYTHON Programming lab	3	4
6	DSC-6: Internet Technologies DSC-6Lab: JAVA Script, HTML, CSS Lab	3	4

* Vocational Courses:

Group-1: <ul style="list-style-type: none"> • DTP, CAD and Multimedia • Hardware and Server Maintenance • Web Content Management Systems • E-Commerce • Web Designing 	Group-1: <ul style="list-style-type: none"> • Health Care Technologies • Digital Marketing • Office Automation • Multimedia Processing • Accounting Package
---	---

** Discipline Specific Elective Courses:

Group-1: <ul style="list-style-type: none"> • IoT • Cyber Law and Cyber Security • Web Programming - PHP and MySQL • Clouds, Grids, and Clusters • Software Testing 	Group-2: <ul style="list-style-type: none"> • Information and Network Security • Data Compression • Discrete Structures • Opensource Programming • Multimedia Computing • Big Data 	Group-3: <ul style="list-style-type: none"> • Data Analytics • Storage Area Networks • Pattern Recognition • Digital Image Processing • Parallel Programming • Digital Signal Processing
---	---	---

Model Syllabus for BSc (Basic and Honors), Semesters I and II

Semester: I

Course Code: DSC-1	Course Title: Computer Fundamentals and Programming in C
Course Credits: 04	Hour of Teaching/Week: 04
Total Contact Hours: 52	Formative Assessment Marks: 30
Exam Marks: 70	Exam Duration: 03

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 and the importance of operating systems
- Understand programming languages, number systems, peripheral devices, networking, multimedia and internet concepts
- 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

Course Content

Content	Hours
Unit - 1	
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.	8
Unit - 2	
Introduction to C Programming: Over View of C; History and Features of C; Structure of a C Program with Examples; Creating and Executing a C Program; Compilation process in C. C Programming Basic Concepts: C Character Set; C tokens - keywords, identifiers, constants, and variables; Data types; Declaration & initialization of variables; Symbolic constants. Input and output with C: Formatted I/O functions - <i>printf</i> and <i>scanf</i> , control stings and escape sequences, output specifications with <i>printf</i> functions; Unformatted I/O functions to read and display single character and a string - <i>getchar</i> , <i>putchar</i> , <i>gets</i> and <i>puts</i> functions.	10
Unit - 3	
C Operators & Expressions: Arithmetic operators; Relational operators; Logical operators; Assignment operators; Increment & Decrement operators; Bitwise operators; Conditional operator; Special operators; Operator Precedence and Associativity; Evaluation of arithmetic expressions; Type conversion.	12

Control Structures: Decision making Statements - <i>Simple if, if_else, nested if_else, else_if ladder, Switch-case, goto, break & continue</i> statements; Looping Statements - Entry controlled and Exit controlled statements, <i>while, do-while, for</i> loops, Nested loops.	
Unit - 4	
Arrays: One Dimensional arrays - Declaration, Initialization and Memory representation; Two Dimensional arrays - Declaration, Initialization and Memory representation. Strings: Declaring & Initializing string variables; String handling functions - <i>strlen, strcmp, strcpy and strcat</i> ; Character handling functions - <i>toascii, toupper, tolower, isalpha, isnumeric</i> etc. Pointers in C: Understanding pointers - Declaring and initializing pointers, accessing address and value of variables using pointers; Pointers and Arrays; Pointer Arithmetic; Advantages and disadvantages of using pointers;	12
Unit - 5	
User Defined Functions: Need for user defined functions; Format of C user defined functions; Components of user defined functions - return type, name, parameter list, function body, return statement and function call; Categories of user defined functions - With and without parameters and return type. User defined data types: Structures - Structure Definition, Advantages of Structure, declaring structure variables, accessing structure members, Structure members initialization, comparing structure variables, Array of Structures; Unions - Union definition; difference between Structures and Unions.	10

Text Books

1. Pradeep K. Sinha and Priti Sinha: Computer Fundamentals (Sixth Edition), BPB Publication
2. E. Balgurusamy: Programming in ANSI C (TMH)

References

1. Kamthane: Programming with ANSI and TURBO C (Pearson Education)
2. V. Rajaraman: Programming in C (PHI – EEE)
3. S. ByronGottfried: Programming with C (TMH)
4. Kernighan & Ritche: The C Programming Language (PHI)
5. Yashwant Kanitkar: Let us C
6. P.B. Kottur: Programming in C (Sapna Book House)

Course Code: DSC-1Lab	Course Title: C Programming Lab
Course Credits: 02	Hour of Teaching/Week: 04
Total Contact Hours: 52	Formative Assessment Marks: 10
Exam Marks: 40	Exam Duration: 04

Practice Lab

The following activities be carried out/ discussed in the lab during the initial period of the semester.

1. Basic Computer Proficiency
 - a. Familiarization of Computer Hardware Parts
 - b. Basic Computer Operations and Maintenance.
 - c. Do's and Don'ts, Safety Guidelines in Computer Lab
2. Familiarization of Basic Software – Operating System, Word Processors, Internet Browsers, Integrated Development Environment (IDE) with Examples.
3. Type Program Code, Debug and Compile basic programs covering C Programming fundamentals discussed during theory classes.

Programming Lab

Part A:

1. Write a C Program to read radius of a circle and to find area and circumference
2. Write a C Program to read three numbers and find the biggest of three
3. Write a C Program to demonstrate library functions in *math.h*
4. Write a C Program to check for prime
5. Write a C Program to generate n primes
6. Write a C Program to read a number, find the sum of the digits, reverse the number and check it for palindrome
7. Write a C Program to read numbers from keyboard continuously till the user presses 999 and to find the sum of only positive numbers
8. Write a C Program to read percentage of marks and to display appropriate message (Demonstration of else-if ladder)
9. Write a C Program to find the roots of quadratic equation (demonstration of switch-case statement)
10. Write a C program to read marks scored by n students and find the average of marks (Demonstration of single dimensional array)
11. Write a C Program to remove Duplicate Element in a single dimensional Array
12. Program to perform addition and subtraction of Matrices

Part B:

1. Write a C Program to find the length of a string without using built in function
2. Write a C Program to demonstrate string functions.
3. Write a C Program to demonstrate pointers in C
4. Write a C Program to check a number for prime by defining *isprime()* function
5. Write a C Program to read, display and to find the trace of a square matrix
6. Write a C Program to read, display and add two m x n matrices using functions
7. Write a C Program to read, display and multiply two m x n matrices using functions
8. Write a C Program to read a string and to find the number of alphabets, digits, vowels, consonants, spaces and special characters.
9. Write a C Program to Reverse a String using Pointer
10. Write a C Program to Swap Two Numbers using Pointers
11. Write a C Program to demonstrate student structure to read & display records of n students.
12. Write a C Program to demonstrate the difference between structure & union.

Note: Student has to execute a minimum of 10 programs in each part to complete the Lab course

Evaluation Scheme for Lab Examination

Assessment Criteria		Marks
Program – 1 from Part B	Flowchart / Algorithm	02
	Writing the Program	05
	Execution and Formatting	08
Program -2 from Part B	Flowchart/Algorithm	02
	Writing the Program	05
	Execution and Formatting	08
Viva Voice based on C Programming		05
Practical Record		05
Total		40

Semester: II

Course Code: DSC-2	Course Title: Data Structures using C
Course Credits: 04	Hour of Teaching/Week: 04
Total Contact Hours: 52	Formative Assessment Marks: 30
Exam Marks: 70	Exam Duration: 03 Hours

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
- Compare alternative implementations of data structures with respect to performance
- Describe the concept of recursion, give examples of its use
- Discuss the computational efficiency of the principal algorithms for sorting and searching

Course Content

Content	Hours
Unit - 1	
Introduction to data structures: Definition; Types of data structures - Primitive & Non-primitive, Linear and Non-linear; Operations on data structures. Algorithm Specification, Performance Analysis, Performance Measurement Recursion: Definition; Types of recursions; Recursion Technique Examples - Fibonacci numbers, GCD, Binomial coefficient nC_r , Towers of Hanoi; Comparison between iterative and recursive functions.	10
Unit - 2	
Arrays: Basic Concepts – Definition, Declaration, Initialisation, Operations on arrays; Types of arrays; Arrays as abstract data types (ADT); Representation of Linear Arrays in memory; Traversing linear arrays; Inserting and deleting elements; Sorting – Selection sort, Bubble sort, Quick sort, Selection sort, Insertion sort; Searching - Sequential Search, Binary search; Iterative and Recursive searching; Multidimensional arrays; Representation of multidimensional arrays; Sparse matrices.	10
Unit - 3	
Dynamic memory allocation: Static & Dynamic memory allocation; Memory allocation and de-allocation functions - <i>malloc</i> , <i>calloc</i> , <i>realloc</i> and <i>free</i> . Linked list: Basic Concepts – Definition and Representation of linked list, Types of linked lists - Singly linked list, Doubly linked list, Header linked list, Circular linked list; Representation of Linked list in Memory; Operations on Singly linked lists – Traversing, Searching, Insertion, Deletion; Memory allocation; Garbage collection,	12

Unit - 4	
Stacks: Basic Concepts – Definition and Representation of stacks; Operations on stacks; Applications of stacks; Infix, postfix and prefix notations; Conversion from infix to postfix using stack; Evaluation of postfix expression using stack; Application of stack in function calls. Queues: Basic Concepts – Definition and Representation of queues; Types of queues - Simple queues, Circular queues, Double ended queues, Priority queues; Operations on Simple queues;	10
Unit - 5	
Trees: Definition; Tree terminologies –node, root node, parent node, ancestors of a node, siblings, terminal & non-terminal nodes, degree of a node, level, edge, path, depth; Binary tree: Type of binary trees - strict binary tree, complete binary tree, binary search tree and heap tree; Array representation of binary tree. Traversal of binary tree; <i>preorder</i> , <i>inorder</i> and <i>postorder</i> traversal; Reconstruction of a binary tree when any two of the traversals are given.	10

Text Books

1. Satraj Sahani: Fundamentals of Data Structures

References

1. Tanenbaum: Data structures using C (Pearson Education)
2. Kamathane: Introduction to Data structures (Pearson Education)
3. Y. Kanitkar: Data Structures Using C (BPB)
4. Kottur: Data Structure Using C
5. Padma Reddy: Data Structure Using C
6. Sudipa Mukherjee: Data Structures using C – 1000 Problems and Solutions (McGraw Hill Education, 2007)

Course Code: DSC-2Lab	Course Title: Data Structures Lab
Course Credits: 02	Hour of Teaching/Week: 04
Total Contact Hours: 52	Formative Assessment Marks: 10
Exam Marks: 40	Exam Duration: 04

Programming Lab

Part A:

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

Part B:

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

Evaluation Scheme for Lab Examination

Assessment Criteria		Marks
Program – 1 from Part A	Flowchart / Algorithm	02
	Writing the Program	05
	Execution and Formatting	08
Program -2 from Part B	Flowchart/Algorithm	02
	Writing the Program	05
	Execution and Formatting	08
Viva Voice based on C Programming		05
Practical Record		05
Total		40

Open Electives in Computer Science:

(For BA, BSc, BCom, BSW, BBA, BBM students studying Core Courses other than Computer Science/ Computer Applications)

- Office Automation
- C Programming Concepts
- Multimedia Processing
- Python Programming Concepts
- R Programming
- E-Content Development
- E-Commerce
- Web Designing
- Computer Animation
- Accounting Package