BENGALURU CITY UNIVERSITY

CHOICE BASED CREDIT SYSTEM

(as per SEP 2024)

Syllabus for I & II Semester
B.Sc. Computer Science

2024-25

1	Dr. Muralidhara B L Professor, Department of Computer Science Bangalore University	Chairperson
2	Dr. Guru D.S Professor PG Department of Computer Science Mysore Univeristy	Member
3	Dr. Susesha Professor, PG Department of Computer Science Mysore Univeristy	Member
4	Dr. Prabhakar C.J Professor Kuvempu University, Shimogga	Member
5	Dr. Chandrakanth Naikodi Associate Professor Department of Computer Science Davanagere University	Member
6	Dr. Prathibha V Kalburgi Ramaiah College of Arts Science, and Commerce Bangalore	Member
7	Mrs. Amalorpavam Sambram Academi of Management Studies Bangalore	Member
8	Dr. Bhagyawana S Mudigowda Associate Professor Maharani Cluster University, Bangalore	Member

BENGALURU CITY UNIVERSITY

Department of Computer Science and Applications

B.Sc. (Computer Science as an Optional)

AS PER STATE EDUCTION POLICY

Sem	Course/ Paper Code	Title of the Paper	Teaching Hours/ week	Semester End Exam	Internal Assessment	Total Marks	Credits
1	24BSC-CS-1	Problem Solving Technique	3	80	20	100	3
	24BSC-CS-1P	Problem Solving Technique Lab	4	40	10	50	2
2	24BSC-CS-2	Data Structure	3	80	20	100	3
	24BSC-CS-2P	Data Structure Lab	4	40	10	50	2
3	24BSC-CS-3	Database Management System	3	80	20	100	3
	24BSC-CS-3P	Database Management System Lab	4	40	10	50	2
4	24BSC-CS-4	Object Oriented Programming using Java	3	80	20	100	3
	24BSC-CS-4P	Object Oriented Programming using Java Lab	4	40	10	50	2
5	24BSC-CS-5	Computer Networks	4	80	20	100	4
	24BSC-CS-6	Operating Systems	4	80	20	100	4
6	24BSC-CS-7	Artificial Intelligence	4	80	20	100	4
	24BSC-CS-8	Project Work	8	80	20	100	4

Department of Computer Science and Applications

BENGALURU CITY UNIVERSITY, BANGALORE

Program Outcome

PO1	Computational Knowledge	Acquire in-depth computational and mathematical knowledge with an ability to abstract and conceptualise from defined problems and requirements.
PO2	Dynamic Problem-Solving Skill	Identify, formulate, and exhibit strong analytical and dynamic problem-solving skills to address evolving computational challenges.
PO3	Innovative System Analysis and Design/ Development	Design and evaluate solutions for complex problems in Data Science, AI & ML, and Full Stack Development, considering societal, cultural, and environmental factors.
PO4	Investigate complex computing problem	Conduct literature surveys, analyze information, and design experiments using appropriate research methods to derive valid conclusions in relevant domains.
PO5	Use of modern tools/ Adaptive programming proficiency	Select, adapt, and apply modern IT tools and programming languages effectively in Data Science, AI & ML, and Full Stack Development to solve diverse computing challenges.
PO6	Knowledge Optimization	Modify algorithms or software systems to improve efficiency or resource utilization.
PO7	Life Long Continuous learning and Technology Adaptability	Pursue lifelong learning to stay updated with emerging technologies in Data Science, AI & ML, and Full Stack Development for sustained employability.
PO8	Soft skills and collaborative teamwork	Communicate effectively, enhance interpersonal skills, and collaborate in multidisciplinary teams essential for success in professional environments.
PO9	Cyber Security Proficiency	Understand cyber threats, develop secure software, and protect sensitive data in Data Science, AI & ML, and Full Stack Development applications.
PO10	Ethical and Professional Conduct	Adhere to ethical standards and professional practices in Data Science, AI & ML, and Full Stack Development roles and responsibilities.
PO11	Employability	Identify market trends, upgrade skills accordingly, and enhance employability in Data Science, AI & ML, and Full Stack Development careers.
PO12	Innovation and Entrepreneurship	Identify opportunities, innovate, and create value through Data Science, AI & ML, and Full Stack Development projects for personal growth and societal impact.

PROBLEM SOLVING TECHNIQUE

Course Outcomes

Upon successful completion of the course, the student will be able:

CO1 To understand algorithmic strategies for enhancing problem-solving proficiency

CO2 Demonstrate problem solving tools and techniques using C.

CO3 To analyze the given problems and use appropriate algorithms.

CO4 To implement sorting and searching techniques to develop programs.

UNIT –1 12 Hours

Introduction: The Role of Algorithms in computing, Algorithms as a technology, analyzing algorithms, Designing algorithms, Growth of Functions, Asymptotic notation, Standard notations and common functions.

Fundamental Algorithms: Exchanging the values of two variables, Counting, Summation of a set of numbers, Factorial Computation, Generating of Fibonacci sequence, Reversing the digits of an integer, Character to number conversion.

UNIT-II 11 Hours

C Programming: Getting Started, Variables, Operators and Arithmetic expressions. Input and Output: Standard input and output, formatted input and output. Selection statements: Statements and Blocks, If, If-else, if-else-if ladder, nested if, switch. Control Structure: while loop, for loop, do-while loop, break and continue, goto and labels. Pointers and Arrays: Pointers and address, Pointers and function arguments, One Dimensional array, Two-Dimensional array, Multidimensional array, Command line arguments.

UNIT - III 11 Hours

Factoring Methods: Finding the square root of a number, the smallest Divisor of an integer, the greatest common divisor of two integers, computing the prime factor of an integer, raising a number to a large power. Array Techniques: Array order reversal, Array counting, Finding the maximum number in a set, removal of duplicates from an ordered array, partitioning an array, finding the kth smallest element, multiplication of two matrices.

UNIT - IV 11 Hours

Sorting: Sorting by selection, sorting by exchange, sorting by insertion, sorting by diminishing increment, sorting by partitioning. Searching: Linear Search, Binary search, Hash search. Text processing and Pattern searching: Text line length adjustment, keyboard searching in text, text line editing, linear pattern searching.

Text Book

1 R. G. Dromey, “How to Solve it by Computer”, Person Education India, 2008.

2 Brain M. Kernighan and Dennis M. Ritchie, “ The C Programming Language”, 2nd edition, Princeton Hall Software Series, 2012.

3 Thomas H Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, “Introduction to Algorithms”, 3rd Edition, The MIT Press Cambridge, Massachusetts London, England, 2008.

Reference Books

1 E. Balaguruswamy, “Programming In ANSI C”, 4th edition, TMH Publications, 2007

2 Greg Perry and Dean Miller, “C programming Absolute Beginner’s Guide”, 3rd edition, Pearson Education, Inc, 2014.

3 Donald E. Knuth, The Art of Computer Programming”, Volume 2: Seminumerical Algorithms, 3rd Edition, Addison Wesley Longman, 1998.

Course Articulation Matrix: Mapping of Course Outcomes(COs) with Program Outcomes(POs1-12)

Course Outcome(COs)	**Program Outcomes(POs)**											
	1	2	3	4	5	6	7	8	9	10	11	12
CO1	3	3	1	1	1	1	2	1	1	1	2	1
CO2	3	3	3	2	1	1	1	1	1	1	2	1
CO3	3	3	1	1	1	1	1	1	1	1	2	1
CO4	3	3	2	2	1	1	1	1	1	1	2	1

Pedagogy: Lecture with the use of ICT/ Field Study / Assignment

Formative Assessment for Theory	
Assessment Occasion Type	**Marks**
C-1 Sessional Tests	5
C-1 Seminars/ Presentations	5
C-2 Sessional Tests	5
Case Study / Assignment / Project work etc.	5
Total	**20 Marks**
Formative Assessments as per SEP guidelines are compulsory	

PROBLEM SOLVING TECHNIQUE LAB

Write, and execute C Program for the following:

1. To read the radius of the circle and to find area and circumference.
2. To read the numbers and find the biggest of three.
3. To check whether the number is prime or not.
4. To find the root of quadratic equation.
5. To read a number, find the sum of the digits, reverse the number and check it for palindrome.
6. To read the numbers from keyboard continuously till the user presses 999 and to find the sum of only positive numbers.
7. To read percentage of marks and to display appropriate message. If a percentage is 70 and above-Distinction, 60-69 – First Class, 50-59 – Second Class, 40-49 Pass, below 40 – Fail. (Demonstrate of if-else ladder)
8. To simulate a simple calculator with addition, subtraction, multiplication, division and it should display the error message for division of zero using switch case.
9. To read marks scored by n students and find the average of marks (Demonstration of single dimensional array)
10. To remove duplicate elements in a single dimensional array.
11. To find the factorial of a number.
12. To generate Fibonacci series.
13. To design the following pattern using nested for loop:

```
        *
      *   *
    *   *   *
  *   *   *   *
*   *   *   *   *
```

14. To find the length of the string without using built-in function.
15. To demonstrate string functions. (String Length, String Copy, String Concatenate, String Comparison)
16. To read, display and add two n x m matrices using function.

17. To read a string and to find the number of alphabets, digits, vowels, consonants, space and special characters.
18. To swap two numbers using pointers.
19. To demonstrate student structure to read & display records of n students.
20. To demonstrate the difference between structure and union for the following
    Student name (String), Student roll no(integer), Student mark(float)

DATA STRUCTURE

Course Outcome

CO1 Understand basic concepts of data structures.

CO2 Analyzing and exploring various ways of storing data using Array and Linked list.

CO3 Demonstrate stack and queue data structures and their applications

CO4 Analyze and implement various non linear data structures.

UNIT I 11 Hours

Introduction and Overview: Definition, Elementary data organization, Data Structures, data Structures operations, Abstract data types, algorithms complexity, time-space trade off. Preliminaries: Mathematical notations and functions, Algorithmic notations, control structures, Complexity of algorithms, asymptotic notations for complexity of algorithms. Introduction to Strings, Storing String, Character Data Types, String Operations, word processing, Introduction to pattern matching algorithms.

UNIT II 11 Hours

Arrays: Definition, Linear arrays, arrays as ADT, Representation of Linear Arrays in Memory, Traversing Linear arrays, Inserting and deleting, multi-dimensional arrays, Matrices and Sparse matrices, searching and sorting techniques using array.
Linked list: Definition, Representation of Singly Linked List in memory, Traversing a Singly linked list, Searching in 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 11 Hours

Stacks: Definition, Array representation of stacks, Linked representation of stacks, Stack as ADT, Arithmetic Expressions: Polish Notation, Conversion of infix expression to postfix expression, Evaluation of Post fix expression, 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 12 Hours

Binary Trees: Definitions, Tree Search, Traversal of Binary Tree, Tree Sort, Building a Binary Search Tree, Height Balance: AVL Trees, Contiguous Representation of Binary Trees: Heaps, Red Black Tree: Insertion and Deletion, External Searching: B-Trees, Applications of Trees. Graphs: Mathematical Back ground, Computer Representation, Graph Traversal. Hashing: Hash Table ADT, understanding Hashing, Components of Hashing, Hash Table, Hash Function, Hashing Techniques, collisions, collision resolution techniques.

Text Book

1 Seymour Lipschutz, “Data Structures with C”, Schaum’s outLines, Tata Mc Graw Hill, 2011.

2 Robert Kruse, C.L.Tondo, Bruce Leung,Shashi Mogalla,“Data Structures and Program Design using C”, Pearson Education, 2009

Reference Books

1 Mark Allen Weiss,“ Data Structures and Algorithm Analysis in C”, Second Edition, Pearson Education,2013

2 Forouzan,“A Structured Programming Approach using C”,2nd Edition, Cengage LearningIndia,2008.

Course Articulation Matrix: Mapping of Course Outcomes (COs) with Program Outcomes (POs12)

Course Outcome (COs)	**Program Outcomes (POs)**											
	1	2	3	4	5	6	7	8	9	10	11	12
CO1	3	4	3	4	4	4	3	2	3	1	4	4
CO2	5	5	4	4	4	4	3	2	3	1	4	4
CO3	5	5	4	4	4	4	3	2	3	1	4	5
CO4	5	5	4	4	4	4	3	2	2	1	4	4

Pedagogy: Lecture with the use of ICT/ Field Study / Assignment

Formative Assessment for Theory	
Assessment Occasion Type	**Marks**
C-1 Sessional Tests	5
C-1 Seminars/ Presentations	5
C-2 Sessional Tests	5
Case Study / Assignment / Project work etc.	5
Total	**20 Marks**
Formative Assessments as per SEP guidelines are compulsory	

DATA STRUCTURE LAB

NOTE: For all the programs write the output, flowchart and number of basic operations performed.

1. Write a program to search for an element in an array using binary and linear search.
2. Write a program to sort list of n numbers using Bubble Sort algorithms.
3. Perform the Insertion and Selection Sort on the input {75,8,1,16,48,3,7,0} and display the output in descending order.
4. Write a program to insert the elements {61,16,8,27} into singly linked list and delete 8,61,27 from the list. Display your list after each insertion and deletion.
5. Write a program to insert the elements {45, 34, 10, 63,3} into linear queue and delete three elements from the list. Display your list after each insertion and deletion.
6. Write a program to simulate the working of Circular queue using an array.
7. Write a program to insert the elements {61,16,8,27} into ordered singly linked list and delete 8,61,27 from the list. Display your list after each insertion and deletion.
8. Write a program for Tower of Honoi problem using recursion.
9. Write recursive program to find GCD of 3 numbers.
10. Write a program to demonstrate working of stack using linked list.
11. Write a program to convert an infix expression x^y/(5*z)+2 to its postfix expression
12. Write a program to evaluate a postfix expression 5 3+8 2 - *.
13. Write a program to create a binary tree with the elements {18,15,40,50,30,17,41} after creation insert 45 and 19 into tree and delete 15,17 and 41 from tree. Display the tree on each insertion and deletion operation.
14. Write a program to create binary search tree with the elements {2,5,1,3,9,0,6} and perform inorder, preorder and post order traversal.
15. Write a program to Sort the following elements using heap sort {9.16,32,8,4,1,5,8,0}.

16. Given S1={“Flowers”} ; S2={“are beautiful”} I. Find the length of S1 II. Concatenate S1 and S2 III. Extract the substring “low” from S1 IV. Find “are” in S2 and replace it with “is” .

17. Write a program to implement adjacency matrix of a graph.

18. Write a program to insert/retrieve an entry into hash/ from a hash table with open addressing using linear probing.