

CBM-105 Programming in C with Data Structures

Course Title	Programming In C with Data Structures	Course Code	CBM-105
Year/Semester	I Year	Course Group	Core
Teaching Scheme in Periods (L:T:P)	4:1:0	Credits	3
Methodology	Lecture+Tutorial	Total Contact Periods	150
CIE	60 Marks	SEE	40 Marks

Prerequisites

Knowledge of basic mathematics and IT skills.

Upon completion of the course the students shall be able to:

Course Outcome	
CO1	Programming Methodology and Introduction to C Language
CO2	Decision making, Looping statements, Arrays and Strings
CO3	User defined functions, Structures and Unions and File Management
CO4	File Management & Pointers
CO5	Dynamic Memory allocation, Searching & Sorting
CO6	Stacks and Queues, Linked Lists

COURSE CONTENTS:

Unit-1 Programming Methodology and Introduction to C Language

Duration: 20 Periods

Steps involved in problem solving -Define Algorithm - Characteristics of algorithm - Steps involved in algorithm development - Algorithms for simple problems - Define flowchart - Symbols used in flowchart - Flowcharts for simple problems - Differentiate algorithm and flowchart – Define Program - Differentiate between algorithm and program - Define High level language and low level language- Define Compiler, Assembler, Linker and Loader - Define source code, assembly code, object code, machine code -importance of C language -- structure of C program – different stages in converting the C program source code into an executable code.

Character set - C Tokens - Keywords and Identifiers- Constants and Variables - data types and classification - declaration of a variable - Assigning values to variables - Define operator - Classify operators - List various special operators - precedence and associativity of operators - Define expression - Evaluation of expressions - type conversion techniques – Know Preprocessor directives – Know Header Files - Reading and writing characters - formatted input and output.

Unit-2 Decision making, Looping statements, Arrays and Strings

Duration:22 Periods

Decision making in C programming-simple if statement– *if-else* statement– nested if statement

- *else-if* ladder – break and continue statements - Switch statement - Classification of various loop statements- while loop – *do-while* loop - for loop - nesting of loops- Comparisons of different loop statements - Infinite loop and steps to avoid it.

Arrays – definition, declaration and initialization of One Dimensional -Array -Accessing the elements in the Array - ‘accessing array out of bounds’ problem - Reordering an array in ascending order – average of elements in an array – largest element in an array - Declaration and initialization of two Dimensional Arrays - sample programs on matrix addition and matrix multiplication, transpose- Strings – Definition, Declaration and initialization of String variables - Reading of strings from terminal - writing strings to screen - String handling functions with sample programs

Unit-3 User defined functions, Structures and Unions

Duration:20 Periods

Function – Definition - Need for user defined functions - Advantages of functions - elements of function -Return values and their types - function prototype - function call - function definition – Scope, visibility and lifetime of variables in functions- Local and External variables -Global variable-Functions with no arguments and no return values - functions with arguments with no return values - functions with arguments with return values - functions with no arguments with return values - Recursion - sample programs on recursion - passing arrays to functions .

Structures and Unions Structure-Definition-Creating a structure-Declaring structure variables -Accessing the structure members - Concept of structure assignment - Find size of a structure - Nested structure - Structure as function arguments - Array of structures - Structure containing arrays - array of structures containing arrays - Union and illustrate usage of a union

– difference between Structures and Union.

UNIT - 4: File Management & Pointers

Duration: 20 Periods

Files management File – Definition -Declare file pointer to a file – file opening modes - Concept of closing of a file - Input / Output operations on a file - Random access to files.

Pointer- Declaration and Initialization of Pointers -Accessing the address of a variable using & operator- Accessing a value of a variable using de-referencing(*) pointer - Differentiate address and de-referencing operators - Pointer Arithmetic-precedence of address and de-referencing operators-Relationship between Arrays and Pointers- Accessing array elements using pointers-Pointers as Function Arguments-Discuss Array of Pointers with examples-pointers and structures. Types of pointers –generic pointer, NULL pointer, dangling pointer, pointer to pointer

UNIT – 5 Dynamic Memory allocation , Data structures, Sorting and Searching

Duration: 12 Periods

Dynamic Memory allocation Introduction to Dynamic memory allocation, Allocating a block of memory using malloc (), Allocating multiple blocks of memory using calloc (), Releasing the used space using free (), Altering the size of memory using realloc (). Compare and contrast different dynamic memory allocation functions.

Data structures – Abstract data type(ADT), Classification of data structures -Linear & nonlinear Data structures, Algorithm analysis- space and time complexity, Asymptotic Notations (big-oh, Omega, Theta Notation).

Sorting and Searching Sorting Algorithms– Bubble sort – algorithm –program- analysis for time complexity, selection sort – algorithm- program- Analysis for its time complexity, Insertion sort – Algorithm- Program and Algorithm Analysis of time complexity, Divide and conquer strategy-Quick sort –Algorithm-Program and analysis for time complexity& Merge sort –Algorithm- Algorithm analysis for time complexity, Application of sorting techniques. Searching Techniques – Linear Search – algorithm- program – algorithm analysis, Binary Search – algorithm – Program –algorithm analysis, Applications of searching techniques.

UNIT - 6: Stacks and Queues, Linked Lists

Duration: 12 Periods

Stacks – Definition - Array Implementation of stacks, Application of stacks, converting infix to postfix expression – Algorithm, Evaluation of postfix expression – Algorithm –Program.

Queues – Definition- Array implementation of Queue, disadvantage of Ordinary queue,

Circular queue- Array implementation of Circular queue– Sparse matrix.

Linked Lists – Single linked list – routines on - Inserting the node at the beginning, Inserting the node at the end, Inserting the node at any random position, Deleting the node at the beginning, Deleting the node at the end, Deleting the node at any random position –

Doubly linked list- Inserting the node at the beginning, Inserting the node at the end, Inserting the node at any random position, Deleting the node at the beginning , deleting the node at the end, Deleting the node at any random position- Create and display circular singly linked list- Display- program comprising all the functions, Advantages and disadvantages of Single/double linked list.

REFERENCE BOOKS

1. Let Us C -Yashwanth Kanetkar.
2. ANSI C –E Balaguru Swamy
3. Data Structures :APseudo code Approach with C++ -Gilberg/ Forouzan
4. Data Structures using C –AaronM. Tanenbaum.
5. Data structures through C -Yashwanth Kanetkar
6. C The complete Reference - Schildt Tata McGraw Hill

Suggested Learning Outcomes

CO1: Programming Methodology and Introduction to C Language

- 4.1 State different steps involved in problem solving
- 4.2 Define algorithm and know its characteristics
- 4.3 State the steps involved in algorithm development
- 4.4 Develop algorithms for simple problems.
- 4.5 Define flowchart and know symbols used in drawing flowcharts
- 4.6 Draw flowchart for simple problems.
- 4.7 Differentiate algorithm and flowchart
- 4.8 Define program and differentiate between program and algorithm
- 4.9 Define High level language and low level language
- 4.10 Define Compiler, Assembler, Linker and Loader
- 4.11 Define source code, assembly code, object code, machine code.
- 4.12 State the importance of C language
- 4.13 Explain the structure of C program List and Explain different stages in converting the C program source code to executable code.
- 4.14 Explain the steps involved in executing the C program

- 4.15 Describe character set, C Tokens-Keywords, Variables, Identifiers, Constants and Variables.
- 4.16 Define Data type. Classify data types and explain with examples
- 4.17 Explain declaration of a variable and assigning values to variables
- 4.18 Define operator.
- 4.19 Classify and Explain operators with examples
- 4.20 Describe precedence and associativity of operators
- 4.21 Define expression
- 4.22 Describe evaluation of Arithmetic, Relational and logical expressions
- 4.23 Illustrate type conversion techniques.
- 4.24 Define and List Pre-processor directives
- 4.25 Discuss #include and #define Pre-processor directives.
- 4.26 Define Header file and discuss stdio.h, conio.h, math.h, string.h Header files.
- 4.27 Illustrate getchar(), putchar(), scanf(), printf() with programs.

CO2: Decision-making, Looping statements, Arrays and Strings

- 2.1 Discuss decision making in programming and
- 2.2 Explain decision making statements- if , if-else, nested if, else-if ladder - with syntax and programs
- 2.3 Discuss break and continue statements.
- 2.4 Explain decision making statement-switch statement –with syntax and programs
- 2.5 Define looping and list loop statements
- 2.6 Explain the loop statements–while,do-while,for statement-with syntax and programs
- 2.7 Define nesting of loops and implement it
- 2.8 Compare different loop statements
- 2.9 Know about Infinite loop and steps to avoid it
- 2.10 Define Array. Describe declaration and initialization of One Dimensional Array with syntax and program
- 2.11 Explain about ‘accessing array out of bounds’ problem
- 2.12 Explain reading elements of an array, accessing the elements and display them with program
- 2.13 Explain sorting elements in an array in ascending order
- 2.14 Explain finding the average of elements in an array and finding the largest element in an array
- 2.15 Explain declaration and initialization of two Dimensional Arrays
- 2.16 Illustrate the concept of two dimensional arrays with sample programs on matrix

addition, subtraction, matrix multiplication and transpose of a matrix

2.17 Define String and know about declaration and initialization of a String variable.

2.18 Know about reading & writing of strings with programs

2.19 Explain about various String handling functions with sample programs.

CO3: User defined functions, Structures, Unions & Files

3.1 Define function, state the need for user defined function, advantages of functions

3.2 Know the elements of function and return values.

3.3 Define and discuss function proto type, function call, and function definition with syntax and program.

3.4 Discuss the scope, visibility and lifetime of variables in functions

3.5 Differentiate Local and Global variables

3.6 Illustrate functions with no arguments and no return values with programs

3.7 Illustrate functions with no arguments and with return values with programs

3.8 Illustrate functions with arguments and with no return values with programs

3.9 Illustrate functions with arguments and with return values with programs

3.10 Illustrate passing arrays to functions with programs

3.11 Define recursion and Illustrate recursion with programs

3.12 Define structure

3.13 Illustrate creating a structure tag and declaring structure variables,

3.14 Explain accessing structure members and structure assignment

3.15 Know size of a structure.

3.16 Discuss nested structure concept.

3.17 Illustrate array of structures, structures containing arrays, arrays of structures containing arrays.

3.18 Illustrate structure as function arguments and returning of structure variables as function values.

3.19 Define Union and illustrate usage of a union.

3.20 Know the differences between Structure and Union

CO4: Files management & Pointers

4.1 Define file

4.2 Know how to declare file pointer to a file

4.3 Illustrate the concept of file opening in various modes

4.4 Illustrate the concept of closing of a file

4.5 Illustrate the concept of Input/Output operations on a file

4.6 Illustrate the concept of random access to files

4.7 Define Pointer

4.8 Illustrate declaration and initialization of Pointers.

- 4.9 Illustrate accessing the address of a variable using & operator.
- 4.10 Illustrate accessing a value of a variable using dereferencing (*)pointer.
- 4.11 Differentiate between address of operator (&) and de-referencing operators(*)
- 4.12 Discuss about pointer arithmetic.
- 4.13 Illustrate precedence of address of operator (&)and de-referencing operators (*)
- 4.14 Illustrate relationship between arrays and pointers.
- 4.15 Illustrate accessing array elements using pointers.
- 4.16 Discuss Array of Pointers with examples.
- 4.17 Illustrate use of pointer to structure.
- 4.18 Illustrate concept of structures containing pointers.
- 4.19 Explain Self-referential structures with examples.
- 4.20 List out different types of pointers.

CO6: Dynamic Memory Allocation, Sorting & Searching

- 5.1 Explain dynamic memory management functions malloc(), calloc(), free() and realloc () and illustrate with examples to use these functions.
- 5.2 Compare different dynamic memory allocation functions.
- 5.3 Define data structure and classify them.
- 5.4 Define space and time complexity.
- 5.5 Different Asymptotic notations used to analyze the algorithms.
- 5.6 Explain about abstract datatypes
- 5.7 State algorithm analysis for time requirements
- 5.8 Define sorting.
- 5.9 State the need of sorting.
- 5.10 List the four methods of sorting.
- 5.11 Explain the method of bubble sort.
- 5.12 Write the algorithm for bubble sort and define its complexity.
- 5.13 Discuss the program for bubble sort.
- 5.14 Explain the method of selection sort.
- 5.15 Write the algorithm for selection sort and define its complexity.
- 5.16 Discuss the program for selection sort.
- 5.17 Explain the method of insertion sort.
- 5.18 Write the algorithm for insertion sort and define its complexity.
- 5.19 Discuss the program for insertion sort.
- 5.20 Explain the method and program on quicksort
- 5.21 Explain the method of merge sort.
- 5.22 List applications of sorting.
- 5.23 Define searching
- 5.24 Explain the method of Linear Search.
- 5.25 Illustrate the algorithm for Linear Search and its complexity
- 5.26 Discuss the program for Linear Search
- 5.27 Explain the method of Binary Search

- 5.28 Write the algorithm for Binary Search and its complexity
- 5.29 Discuss the program for Binary
- 5.30 Applications of searching and sorting

CO6: Stacks and Queues and Linked Lists

- 6.1 Define stack.
- 6.2 List out the operations of a stack.
- 6.3 Implementation of stacks using arrays.
- 6.4 List the applications of stacks.
- 6.5 Algorithm to Convert in fix expression to postfix expression.
- 6.6 Evaluate post fix expression and program to evaluate postfix expression.
- 6.7 Define queue.
- 6.8 List out the operations on queues.
- 6.9 Implement array implementation of queue.
- 6.10 Disadvantage of Ordinary queue.
- 6.11 Implement of circular queues using arrays.
- 6.12 List out the applications of queues.
- 6.13 Definition of sparse matrix – converting ordinary matrix to sparse matrix.
- 6.14 Inserting the node at the beginning, end and at any random position of a singly linked list.
- 6.15 Deleting the node from the beginning, end and at any random position of a singly linked list.
- 6.16 summarize the different routines of singly linked list along with display operation in the program.
- 6.17 Inserting the node at the beginning, end and at any random position of a doubly linked list.
- 6.18 Deleting the node from the beginning, end and at any random position of a doubly linked list.
- 6.19 summarize the different routines of doubly linked list in the program along with display operation in doubly linked list.
- 6.20 summarize the different routines of singly circular list in the program along with display operation in circular linked list.
- 6.21 Advantages and disadvantage of single, double, circular linked lists.

Student activity like mini-project, surveys, quizzes, etc. should be done in group of 5-10 students.

1. Each group should do any one of the following type of activity or any other similar activity related to the course with prior approval from the course coordinator and programme coordinator concerned.
2. Each group should conduct different activity and no repetition should occur.
3. Explore and analyses topics to improve the level of creativity and analytical skill by taking Quiz/ tests/ assignments. Documents have to be maintained as a record.
4. Create a power point presentation on the topic relevant to course or advanced topic as an extension to the course to improve the communication skills. Documents have to be maintained as a record.
5. Visit different sites relevant to topics. Listen to the lectures and submit a handwritten report
6. Coding competitions

CO-PO Mapping Matrix

Course Outcome		CL	Linked PO	Teaching Hours
CO1	Programming Methodology and Introduction to C Language	R,U, A	1,2,3,4,7	10
CO2	Decision making, Looping statements, Arrays and Strings	R,U, A	1,2,3,4,7	10
CO3	User defined functions, Structures and Unions and File Management	U, A	1,2,3,4,7	14
CO4	File Management & Pointers	U, A	1,2,3,4,7	14
CO5	Dynamic Memory allocation, Searching & Sorting	U, A	1,2,3,4,7	13
CO6	Stacks and Queues, Linked Lists	R,U, A	1,2,3,4,7	14
		Total Sessions		75

MID SEM– I Exam

S.No	Unit Name	R	U	A	Remarks
1	Unit-I	1, 2	5(a) 5(b)	7(a) 7(b)	
2	Unit-II	3, 4	6(a) 6(b)	8(a) 8(b)	
Total Questions		4	4	4	

MID SEM– II Exam

S.No	Unit Name	R	U	A	Remarks
1	Unit-I	1, 2	5(a) 5(b)	7(a) 7(b)	
2	Unit-II	3, 4	6(a) 6(b)	8(a) 8(b)	
Questions		4	4	4	

Semester End Examination

S.No	Unit Name	R		U	A	Remarks
1	Unit-I	4	1	9(a)	13(a)	
2	Unit-II					
3	Unit-III		2	10(a)	14(a)	
4	Unit-IV					
5	Unit-V	3	5,6	9(b) 11(a) 11(b)	13(b) 15(a) 15(b)	
6	Unit-VI		7,8	10(b) 11(a) 11(b)	14(b) 16(a) 16(b)	
Total Questions		8		8	8	

Legend:	Remembering(R)	1 Mark
	Understanding(U)	3 Marks
	Application(A)	5 Marks

MODEL QUESTION PAPER
BOARD DIPLOMA MID-SEM-1 EXAMINATION(C-26)
CBM-105 Programming In C with Data Structures

PART-A

MARKS:4 X1=4

NOTE:1. Answer all questions.

1. Each question carries one mark.

1. List any two differences between algorithm and flowchart.
2. Define Low level language
3. List various types of constants.
4. Define an expression.

PART-B

MARKS:2 X3=6

NOTE:1. Answer any one question from 5 and 6.

2.Each question carries three marks.

5.a) Write any three differences between algorithm and flowchart

(OR)

b) Write any three differences between high level language and low level language

6.a) How to declare variable in C? Give examples.

(OR)

b) Evaluate an expression $(2+5)*(10-4)\%10$.

PART-C

MARKS: 2 X5=10

NOTE:1. Answer any one question from 7 and 8.

2.Each question carries five marks.

2. a) Draw a flowchart to find factorial of given number N.

(OR)

b) Draw a flowchart to find biggest of three numbers.

3. a) Explain different data types in C with examples.

(OR)

b) Explain operators in C with examples.

MODEL QUESTION PAPER
BOARD DIPLOMA MID-SEM-2 EXAMINATION (C-26)
CBM-105 Programming In C with Data Structures

PART-A

MARKS: 4X1=4

NOTE:1. Answer all questions.

2. Each question carries one mark.

1. Give the syntax of if statement in C.
2. Give the syntax of while statement in C.
3. Define an array.
4. Give the syntax for declaring and initializing of one dimensional array.

PART-B

MARKS: 2 X3=6

NOTE:1. Answer any one question from 5 and 6.

2. Each question carries three marks.

5. a) Write any three differences between break and continue statements.
(OR)
5. b) Write any three differences between while and do-while statements.
6. a) Write a C program to find largest number in array.
(OR)
6. b) Explain any three String handling functions.

PART-C

MARKS: 2X5=10

NOTE:1. Answer any one question from 7 and 8.

2. Each question carries five marks.

7. a) Explain do-while statement with syntax and sample program
(OR)
7. b) Explain switch statement with syntax and sample program.
8. a) Write a C program to find the multiplication of two matrices.
(OR)
8. b) Write a C program to find the addition of two matrices.