Course Handout

Institute/School Name	Chitkara University Institute of Engineering and Technology		
Department Name	Department of Computer Science & Engineering		
Programme Name	Bachelor of Engineering (B.E.), Computer Science & Engineering		
Course Name	Fundamentals of C Programming	Session	Jan 2026- Jun-2026
Course Code	24CSE0107	Semester/Batch	2nd/ 2025
L-T-P(Per Week)	2-0-4	Course Credits	4
Pre-requisite	N/A	NHEQF Level	4.5
Course Coordinator	Dr. Gaganpreet Kaur	SDG Number	4, 9

1. Objectives of the Course

C programming is a powerful, general-purpose, high-performance programming language. It is a compiled, statically-typed language known for its efficiency and close interaction with hardware. C is widely used in system programming, embedded systems, and application development. Its portability allows code to run on various platforms, making it a preferred choice for building operating systems and other performance-critical applications. This course provides a wide scope of learning & understanding of C programming. The main objectives of the course are:

- To impart knowledge about the different problem solving aspects including general problem solving strategies and working backwards from solution.
- To expose students to the concepts like variables, identifiers, data types, basic input/output, operators etc. for C-Language
- To provide skills to use different control statement (sequential, conditional and iterative), the concepts of pointers and functions for logic building.
- To enable learners to assess the lifecycles of different identifiers by providing the knowledge about different storage structures and array implementations.
- To create efficient programming solutions in common engineering design situations.

2. Course Learning Outcomes (CLOs)

	CLOs	Program Outcomes (PO)	NHEQF Level Descriptor	No. of Lectures
CLO01	Understand C-Language features and basics of problem solving aspects for logic building.	PO1, PO2, PO3, PO5, PO12	Q1, Q3, Q6	4
CLO02	Use of variables, data types, identifiers, different operators and expressions	PO1, PO3, PO4, PO5	Q1, Q2	14
CLO03	Apply conditional statements, switch case statements and iterative statements as flow controls in C-Language to solve complex problems.	PO1, PO2, PO3, PO4, PO5, PO7, PO11	Q3, Q6	11
CLO04	Implement functions, recursion to solve complex problems and observe the use of storage classes in C-language.	PO3, PO4, PO5, PO12	Q3	20
CLO05	Use pointers and one/two dimensional arrays to store and retrieve data items in C-language.	PO4, PO5	Q1, Q2	25
CLO06 (Only for lab component)	Develop real-world applications using the C Programming concepts.	PO1, PO2, PO3, PO4, PO5, PO7, PO11, PO12	Q3, Q6	16
Total Contact Hours				**90**

CLO-PO Mapping

CLO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	Type of Assessment's
CLO01	H	M	M		H							M	Formative,Summative
CLO02	M		M	H	M								Formative,Summative
CLO03	L	M	H	M	M		L				L		Formative,Summative
CLO04			L	M	H							M	Formative,Summative
CLO05				L	H								Formative,Summative
CLO06	L	M	H	M	M		L				H	M	Formative

CLO	PSO1	PSO2
CLO01	H	M
CLO02	H	M
CLO03	H	M
CLO04	H	H
CLO05	M	H
CLO06	H	H

H=High, M=Medium, L=Low

3. Recommended Books:

B01: Kanetkar, Yashwant, "Let us C", New Delhi BPB Publication 2019. 17th Edition.
B02: E. Balagurusamy,"Programming in ANSI C", McGraw Hill Education India, 2019, Edition-8
B03: Reema Thareja, " Computer Fundamentals and Programming in C", Oxford University Press, 2016, 2nd Edition
B04: Schildt, Herbert, "C: The Complete Reference", McGraw Hill Education (New Delhi), 2018, 4th Edition
B05: Kernighan,Brian W.and Ritchie,Dennis M, "The C Programming Language", Pearson Education (New Delhi), 2007, 2nd Edition
B06: E. Balagurusamy,"Fundamentals of Computers", McGraw Hill Education India, 2011, Edition-6

4. Other readings and relevant websites:

Serial No	Link of Journals, Magazines, websites and Research Papers
1.	https://nptel.ac.in/courses/106106210
2.	https://www.coursera.org/specializations/c-programming
3.	https://www.coursera.org/learn/c-for-everyone
4.	https://ocw.mit.edu/courses/6-087-practical-programming-in-c-january-iap-2010/
5.	https://ocw.mit.edu/courses/6-s096-introduction-to-c-and-c-january-iap-2013/
6.	https://catalogue.library.cern/literature/t50vw-6at59

5. Recommended Tools and Platforms

GCC Compiler (GNU Compiler Collection), Code::Blocks IDE, Dev-C++, Visual Studio Code (with C/C++ extensions), Replit (C environment)

6. Course Plan: Theory+ Lab

Theory Plan

Lect. No.	Topic(s)
1-2	History of C, understanding its role in system programming and as a foundation for other languages, Preprocessor directives, the main() function, comments, and basic syntax.
3-4	Input and Output- Using printf() for output and scanf() for input Variables and Constants- Declaration, initialization, and naming conventions.
5-6	Data Types-Basic types: int, char, float, double, Modifiers: signed, unsigned, short, long.
7-8	Operators-Arithmetic, relational, and logical operators, Assignment, increment/decrement, and bitwise operators
9-10	Ternary operator, Type Conversion- Implicit and explicit casting

11-13	Statements- if, if-else, else-if ladder, and switch-case Loops (Iteration): for, while, and do-while loops; Jump Statements—break, continue, and goto
14-16	Arrays- Declaration, initialization, and accessing elements of one-dimensional arrays, Multidimensional arrays (2D arrays).
17-19	Strings: Introduction to strings as character arrays, String manipulation using standard library functions (strcpy, strcat, strlen, strcmp, etc.) from <string.h>.
ST1 (Lect. No. 1-19)	
20-23	Functions and Recursion: Definition, declaration (prototype), and calling functions Types of Functions- User-defined vs. standard library functions, Parameter Passing- Call by value vs. call by reference.
24-25	Recursion- Understanding recursive functions and their applications (e.g., factorial, Fibonacci series).Storage Classes- auto, extern, static, and register.
26-27	Pointers: Understanding memory addresses, declaration, and initialization of pointers, Pointer Operators- Address-of operator (&) and dereference operator (*).
28	Pointer Arithmetic- Incrementing and decrementing pointers, Relationship between arrays and pointers, pointer to an array, and array of pointers,
29-30	Passing pointers to functions and returning pointers from functions, Function Pointers: Pointers that point to functions.
31-32	Dynamic Memory Allocation: Memory Management Functions-using malloc(), calloc(), realloc(), and free() from <stdlib.h>, Memory Leaks: Understanding and avoiding memory leaks
33-34	User-Defined Data Types: Structures- Defining, declaring, and accessing structure members. Arrays of structures and pointers to structures
35-36	Unions- Defining and using unions, understanding the memory difference between structures and unions
37	Enumerations- Creating and using enumerated data types for named integer constants. typedef- Creating aliases for existing data types.
ST2 (Lect. No. 20-37)	
38-39	File Operations: Opening, closing, reading, and writing files, File Modes- Understanding different file access modes (r, w, a, r+, w+, a+)
40-41	File I/O Functions-fopen(), fclose(), fprintf(), fscanf(), fgetc(), fputc(), fgets(), fputs(), fread(), fwrite(), Random Access to Files- Using fseek(), ftell(), and rewind().
42-43	Preprocessor Directives- #include, #define, #if, #else, #endif, macros, Command Line Arguments- Passing arguments to the main() function (argc and argv)
44-45	Bit Fields- Using bit fields inside structures to save memory.Variable Argument Lists- Using <stdarg.h> to create functions that accept a variable number of arguments.
End Term Exam (1-45)	

Lab Plan

Lab No.	Topic(s)
1–2	Introduction to C: Install GCC/Code::Blocks, write and execute the first "Hello, World!" program. Understand the structure of a C program, the main() function, comments, and basic syntax. Discuss the history and relevance of C in system programming and its influence on other languages. Practice with simple preprocessors (#include) and compilation flow.
3–4	Input and Output in C: Practice using printf() and scanf() for user interactions. Work with variables and constants—declaration, initialization, and naming conventions. Create programs using user input/output for different data types.
5–6	Explore data types: int, char, float, double and type modifiers (short, long, signed, unsigned). Write example programs demonstrating valid conversions between types.
7–8	Operators in C: Practice arithmetic, relational, logical, assignment, increment/decrement, and bitwise operators. Work on exercises that illustrate precedence and associativity.
9–10	Ternary operator and Type Conversion: Implement implicit and explicit type casting in calculations. Create programs for conditional value assignment using the ternary operator.
11–13	Control Statements and Iteration: Work with if, else if, switch, and goto. Practice loops (for, while, do-while) in repetitive tasks. Demonstrate break and continue in control flow.
14–16	Arrays: Declare and use one-dimensional and two-dimensional arrays. Write programs for element storage, accessing, and performing mathematical operations (e.g., sum, average, matrix operations).
17–19	Strings: Work with strings as character arrays. Practice standard library string functions (strcpy, strcat, strlen, strcmp, etc.) and manual string manipulation like reversing a string, counting vowels, or detecting palindromes.
20–23	Functions and Recursion Basics: Define, declare, and call user-defined functions. Demonstrate call by value and call by reference. Develop small modular programs showcasing function decomposition.
24–25	Deep dive into Recursion: Write recursive programs like factorial, Fibonacci series.Explore and demonstrate storage classes

	(auto, static, extern, register) through examples.
26–27	Pointers Fundamentals: Practice declaring, initializing, and using pointers. Understand address-of (&) and dereference (*) operators with simple examples.
28	Pointer Arithmetic and Arrays: Demonstrate incrementing and decrementing pointers, pointer to array, and array of pointers through practical examples.
29–30	Functions with Pointers: Pass pointers to functions, return pointers, and understand function pointers. Create interactive examples showing pointer–function relationships.
31–32	Dynamic Memory Allocation: Use malloc(), calloc(), realloc(), and free() for runtime memory management. Diagnose and avoid memory leaks through structured examples.
33–34	Structures: Define and use structures. Access structure members and work with arrays of structures. Use pointers to structures for dynamic data handling.
35–36	Unions and Comparison with Structures: Create programs showing union memory sharing. Understand memory efficiency differences between unions and structures.
37	Enumerations and typedef: Create named constant sets using enum. Use typedef to simplify data type definitions in structured programs.
38–39	File Operations: Practice opening, closing, reading, and writing files. Understand and apply file modes (r, w, a, r+, w+, a+). Create programs for copying file contents.
40–41	File I/O Functions: Implement file reading/writing using functions such as fprintf(), fscanf(), fgetc(), fputc(), fgets(), fputs(), fread(), fwrite(). Practice random access in files using fseek(), ftell(), and rewind().
42–43	Preprocessor Directives and Command-Line Arguments: Use #define, #if, #else, #endif and macros in code. Learn to pass and handle command-line arguments (argc, argv) through mini CLI-based tasks.
44–45	Bit Fields and Variable Arguments: Create structures with bit fields for memory optimization. Implement variable argument functions using <stdarg.h> (e.g., a function like sum(int count, ...)). Demonstrate practical system-level coding use cases.

7. Delivery/Instructional Resources

Theory Plan:

Lect. No.	Topics	CLO	Book No, CH No, Page No	TLM[7]	ALM[8]	Web References	Audio- Video
1-2	History of C, role in system programming, foundation for other languages, preprocessor directives, main() function, comments, basic syntax	CLO01	B01, CH 1, pp. 1-9; B06, CH 1, pp. 1-15	Lecture	Quiz	https://en.cppreference.com/w/c/preprocessor https://ocw.mit.edu/courses/6-087-practical-programming-in-c-january-iap-2010/resources/mit6_087iap10_lec01/ https://www.geeksforgeeks.org/c-language-introduction/	https://www.youtube.com/watch?v=vydldXpA6ec&list=PLyqSpQzTE6M8O9Oy9t-yhiAUXOi-rmTp_&index=1 https://www.youtube.com/watch?v=SnzvWUFxcYU&list=PLyqSpQzTE6M8O9Oy9t-yhiAUXOi-rmTp_&index=2
3-4	Input/Output using printf()/scanf(); Variables & Constants: declaration, initialization, naming conventions	CLO02	B01, CH 1, pp. 6-18; B02, CH 2, pp. 20-35	Lecture	Test questions	https://www.programiz.com/c-programming/c-input-output https://en.cppreference.com/w/cpp/keywords.html https://www.learn-c.org/	https://www.youtube.com/playlist?list=PL_c9BZzLwBRKKqOc9TJz1pP0ASrxLMtp2 https://nptel.ac.in/courses/106104128?utm
5-6	Data Types: int, char, float, double; Modifiers: signed, unsigned, short, long	CLO02	B01, CH 2, pp. 21-32; B03, CH 3, pp. 45-60	Lecture	Quiz	https://en.cppreference.com/w/c/language/type https://www.geeksforgeeks.org/variables-in-c/ https://randu.org/tutorials/c/basic.php	https://www.youtube.com/playlist?list=PL_c9BZzLwBRKKqOc9TJz1pP0ASrxLMtp2 https://nptel.ac.in/courses/106104128?utm
7-8	Operators: Arithmetic, relational, logical, assignment, increment/decrement, bitwise operators	CLO02	B01, CH 2, pp. 32-40; B02, CH 4, pp. 65-80	Lecture	Brainstorming	Lecture 2 | Practical Programming in C | Electrical Engineering and Computer Science | MIT OpenCourseWare https://en.cppreference.	https://nptel.ac.in/courses/106104128?utm https://www.youtube.com/playlist?list=PL_c9BZzLwBRKKqOc9TJz1pP0ASrxLMtp2

						com/w/cpp/language/expressions.html https://www.learn-c.org/ https://randu.org/tutorials/c/basic.php	
9-10	Ternary operator; Type Conversion: implicit & explicit casting	CLO03	B01, CH 4, pp. 57-66; B02, CH 5, pp. 85-95	Lecture	Quiz, puzzles	Lecture 2 | Practical Programming in C | Electrical Engineering and Computer Science | MIT OpenCourseWare	https://nptel.ac.in/courses/106104128?utm https://www.youtube.com/playlist?list=PL_c9BZzLwBRKKqOc9TJz1pP0ASrxLMtp2
11-13	Statements: if, if-else, else-if ladder, switch-case; Loops: for, while, do-while; Jump: break, continue, goto	CLO03	B01, CH 3-6, pp. 39-104; B02, CH 6-7, pp. 100-140	Lecture	Peer review	Lecture 3 | Practical Programming in C | Electrical Engineering and Computer Science | MIT OpenCourseWare https://en.cppreference.com/w/cpp/language/statements.html https://randu.org/tutorials/c/index.php	https://nptel.ac.in/courses/106104128?utm https://www.youtube.com/playlist?list=PL_c9BZzLwBRKKqOc9TJz1pP0ASrxLMtp2
14-16	Arrays: 1D declaration, initialization, access; Multidimensional (2D) arrays	CLO04	B01, CH 13-14, pp. 239-274; B02, CH 8, pp. 150-175	Lecture	Test questions	https://ocw.mit.edu/courses/6-087-practical-programming-in-c-january-iap-2010/ https://www.learn-c.org/ https://randu.org/tutorials/c/index.php	https://nptel.ac.in/courses/106104128?utm https://www.youtube.com/playlist?list=PL_c9BZzLwBRKKqOc9TJz1pP0ASrxLMtp2
17-19	Strings: character arrays; <string.h> functions: strcpy, strcat, strlen, strcmp etc.	CLO04	B01, CH 15, pp. 296-310; B02, CH 9, pp. 180-200	Lecture	Quiz	https://ocw.mit.edu/courses/6-087-practical-programming-in-c-january-iap-2010/resources/mit6_087iap10_lec05/ https://randu.org/tutorials/c/strings.php	https://nptel.ac.in/courses/106104128?utm https://www.youtube.com/playlist?list=PL_c9BZzLwBRKKqOc9TJz1pP0ASrxLMtp2
20-23	Functions: definition, declaration (prototype), calling; User-defined vs library functions; Parameter passing: call-by-value vs call-by-reference	CLO04	B01, CH 7, pp. 135-149; B02, CH 5, pp. 90-120	Lecture	Test questions	https://www.learn-c.org/ https://www.geeksforgeeks.org/c/c-functions/ https://randu.org/tutorials/c/basic2.php	https://nptel.ac.in/courses/106104128?utm https://www.youtube.com/playlist?list=PL_c9BZzLwBRKKqOc9TJz1pP0ASrxLMtp2
24-25	Recursion: factorial, Fibonacci; Storage Classes: auto, extern, static, register	CLO04	B01, CH 10-11, pp. 173-198; B02, CH 10, pp. 220-240	Lecture	Brainstorming	https://www.sanfoundry.com/recursion-in-c/ https://www.wscubetech.com/resources/c-programming/recursion	https://www.youtube.com/playlist?list=PL_c9BZzLwBRKKqOc9TJz1pP0ASrxLMtp2
26-27	Pointers: memory addresses, declaration, initialization; Operators: & (address-of), * (dereference)	CLO05	B01, CH 9, pp. 157-167; B04, CH 11, pp. 300-320	Lecture	Quiz	https://ocw.mit.edu/courses/6-087-practical-programming-in-c-january-iap-2010/resources/mit6_087iap10_lec05/ https://www.geeksforgeeks.org/c/c-pointers/	https://www.youtube.com/playlist?list=PL_c9BZzLwBRKKqOc9TJz1pP0ASrxLMtp2

28	Pointer Arithmetic: increment/decrement; Arrays & pointers relationship; pointer-to-array, array-of-pointers	CLO05	B01, CH 13, pp. 247-255; B04, CH 12, pp. 330-345	Lecture	Test questions	https://ocw.mit.edu/courses/6-087-practical-programming-in-c-january-iap-2010/resources/mit6_087iap10_lec05/ https://ocw.mit.edu/courses/6-087-practical-programming-in-c-january-iap-2010/resources/mit6_087iap10_lec07/	https://nptel.ac.in/courses/106104128?utm https://www.youtube.com/playlist?list=PL_c9BZzLwBRKKqOc9TJz1pP0ASrxLMtp2
29-30	Pointers in functions: passing/returning pointers; Function pointers	CLO05	B01, CH 7+9, pp. 135-167; B04, CH 13, pp. 350-370	Lecture	Quiz	https://www.learn-c.org/ https://www.geeksforgeeks.org/c/function-pointer-in-c/	https://nptel.ac.in/courses/106104128?utm https://www.youtube.com/playlist?list=PL_c9BZzLwBRKKqOc9TJz1pP0ASrxLMtp2
31-32	Dynamic Memory: malloc(), calloc(), realloc(), free(); Memory leaks prevention	CLO05	B01, CH 14, pp. 267-274; B04, CH 14, pp. 380-400	Lecture	Brainstorming	https://ocw.mit.edu/courses/6-087-practical-programming-in-c-january-iap-2010/resources/mit6_087iap10_lec11/ https://www.geeksforgeeks.org/c/dynamic-memory-allocation-in-c-using-malloc-calloc-free-and-realloc/ https://www.geeksforgeeks.org/c/what-is-memory-leak-how-can-we-avoid/ https://randu.org/tutorials/c/dynamic.php	https://nptel.ac.in/courses/106104128 https://www.youtube.com/playlist?list=PL_c9BZzLwBRKKqOc9TJz1pP0ASrxLMtp2
33-34	Structures: definition, declaration, member access; Arrays of structures; Pointers to structures	CLO05	B01, CH 16, pp. 334-342; B02, CH 11, pp. 250-270	Lecture	Test questions	https://ocw.mit.edu/courses/6-087-practical-programming-in-c-january-iap-2010/resources/mit6_087iap10_lec06/ https://randu.org/tutorials/c/structs.php	https://nptel.ac.in/courses/106104128 https://onlinecourses.swayam2.ac.in/imb26_mg57/preview
35-36	Unions: definition, usage; Memory differences vs structures	CLO05	B01, CH 16, pp. 342-348; B04, CH 15, pp. 410-420	Lecture	Quiz	https://ocw.mit.edu/courses/6-087-practical-programming-in-c-january-iap-2010/resources/mit6_087iap10_lec06/ https://www.geeksforgeeks.org/c/c-unions/	https://nptel.ac.in/courses/106104128?utm https://www.youtube.com/playlist?list=PL_c9BZzLwBRKKqOc9TJz1pP0ASrxLMtp2
37	Enumerations: named constants; typedef: type aliases	CLO05	B01, CH 16, pp. 342-348; B02, CH 11, pp. 270-280	Lecture	Quiz	https://www.learn-c.org/ https://www.geeksforgeeks.org/c/enumeration-enum-c/	https://nptel.ac.in/courses/106104128?utm https://www.youtube.com/playlist?list=PL_c9BZzLwBRKKqOc9TJz1pP0ASrxLMtp2
38-39	File Operations: opening/closing files; Modes: r, w, a, r+, w+, a+	CLO05	B01, CH 17, pp. 400-420; B02, CH 12, pp. 290-310	Lecture	Test questions	https://ocw.mit.edu/courses/6-087-practical-programming-in-c-january-iap-2010/	https://nptel.ac.in/courses/106104128?utm https://www.youtube.com/playlist?list=PL_c9BZzLwBRKKqOc9TJz1pP0ASrxLMtp2

40-41	File I/O: fopen(), fclose(), fprintf(), fscanf(), fgetc(), fputc(), fgets(), fputs(), fread(), fwrite(); Random access: fseek(), ftell(), rewind()	CLO05	B01, CH 17, pp. 420-450; B04, CH 16, pp. 430-460	Lecture	Quiz	https://www.learn-c.org/ https://www.geeksforgeeks.org/c/basics-file-handling-c/ https://www.includehelp.com/c-programs/file-handling-in-c-solved-programs.aspx	https://www.youtube.com/playlist?list=PL_c9BZzLwBRKKqOc9TJz1pP0ASrxLMtp2
42-43	Preprocessor: #include, #define, #if, #else, #endif, macros; Command line arguments: argc, argv	CLO05	B01, CH 12, pp. 363-380; B04, CH 18, pp. 470-490	Lecture	Brainstorming	https://ocw.mit.edu/courses/6-087-practical-programming-in-c-january-iap-2010/	https://www.youtube.com/playlist?list=PLyqSpQzTE6M8O9Oy9t-yhiAUXOi-rmTp_
44-45	Bit Fields in structures; Variable Arguments: <stdarg.h> for functions like sum(int count, ...)	CLO05	B04, CH 15, pp. 420-430; B05, CH 7, pp. 150-160	Lecture	Test questions	https://ocw.mit.edu/courses/6-087-practical-programming-in-c-january-iap-2010/resources/mit6_087iap10_lec06/ https://www.tutorialspoint.com/cprogramming/c_command_line_arguments.htm	https://nptel.ac.in/courses/106104128 https://www.youtube.com/playlist?list=PLyqSpQzTE6M8O9Oy9t-yhiAUXOi-rmTp_

Lab Plan:

Lab No.	Experiment	CLO	TLM[7]	ALM[8]	Web References	Audio-Video
1–2	Install C compiler (GCC/Code::Blocks), set up IDE, compile and run the first "Hello, World!" program. Understand program structure, preprocessors, and compilation flow.	CLO01	Demonstration method using simulation tool	Quiz, Group discussion	https://www.gnu.org/software/gcc/ https://www.codeblocks.org/downloads/ https://www.javatpoint.com/compile-run-c-program https://www.tutorialspoint.com/cprogramming/c_overview.htm https://www.geeksforgeeks.org/compiling-a-c-program-behind-the-scenes/	https://www.youtube.com/watch?v=GWJqsmitR2I https://www.youtube.com/watch?v=KJgsSFOSQv0 https://www.youtube.com/watch?v=8PopR3x-VMY
3–4	Work with variables, identifiers, and constants. Declare and initialize different data types. Practice basic input/output using scanf() and printf().	CLO02	Demonstration method using simulation tool	Quiz, Group discussion	https://www.geeksforgeeks.org/c-variables/ https://www.javatpoint.com/c-variables https://www.tutorialspoint.com/cprogramming/c_input_output.htm https://en.cppreference.com/w/c/language/object	https://www.youtube.com/watch?v=KJgsSFOSQv0&t=1200s https://www.youtube.com/watch?v=ZSPZob_1TOk
5–6	Explore data types (int, char, float, double) and type modifiers. Demonstrate type conversion through example programs.	CLO02	Discussion	Quiz, Flowcharts, PPTs	https://www.geeksforgeeks.org/data-types-in-c/ https://www.javatpoint.com/c-data-types https://www.programiz.com/c-programming/type-conversion https://en.cppreference.com/w/c/language/con	https://www.youtube.com/watch?v=CnHXFqQjbyI&list=PLm8dSOaqLPHI65eHiBKtNrbgV47hGbOaM&index=8

					version https://www.includehelp.com/c/default.aspx?utm_source	
7–8	Implement arithmetic, relational, logical, assignment, increment/decrement, and bitwise operators. Solve precedence-based problems.	CLO02	Lecture	Test questions	https://www.geeksforgeeks.org/operators-in-c/ https://www.javatpoint.com/c-operators https://www.tutorialspoint.com/cprogramming/c_operators.htm https://en.cppreference.com/w/c/language/operator_precedence	https://www.youtube.com/watch?v=371vBlwtm2k&list=PLm8dSOaqLPHI65eHiBKtNrbgV47hGbOaM&index=25
9–10	Use ternary operator and perform implicit and explicit type casting. Develop programs for conditional value assignment.	CLO02	Lecture	Test questions	https://www.geeksforgeeks.org/conditional-or-ternary-operator-in-c/ https://www.programiz.com/c-programming/c-conditional-operator https://www.javatpoint.com/c-type-casting	https://www.youtube.com/watch?v=ZSPZob_1TOk&t=2100s
11–13	Implement conditional statements (if, else if, switch, goto) and loops (for, while, do-while). Demonstrate break and continue.	CLO03	Demonstration method using simulation tool	Quiz, Group discussion	https://www.geeksforgeeks.org/decision-making-c-cpp/ https://www.javatpoint.com/c-if-else https://www.programiz.com/c-programming/c-for-loop https://www.tutorialspoint.com/cprogramming/c_loops.htm	https://www.youtube.com/watch?v=ez14U11KqkI&list=PLm8dSOaqLPHI65eHiBKtNrbgV47hGbOaM&index=9&pp=iAQB https://www.youtube.com/watch?v=B9pox16p9To&list=PLm8dSOaqLPHI65eHiBKtNrbgV47hGbOaM&index=11
14–16	Declare and use one-dimensional and two-dimensional arrays. Perform operations such as sum, average, and matrix calculations.	CLO03	Demonstration method using simulation tool	Quiz, Group discussion	https://www.geeksforgeeks.org/arrays-in-c/ https://www.programiz.com /c-programming/c-arrays https://www.javatpoint.com/c-array https://en.cppreference.com/w/c/language/array https://www.programiz.com/c-programming/c-pointer-examples	https://www.youtube.com/watch?v=T6eSt06zuaA&list=PLm8dSOaqLPHI65eHiBKtNrbgV47hGbOaM&index=12
17–19	Work with strings using standard library functions and manual string manipulation techniques.	CLO06	Discussion	Test questions	https://www.geeksforgeeks.org/strings-in-c/ https://www.javatpoint.com/c-string https://www.programiz.com/c-programming/c-strings https://en.cppreference.com/w/c/string/byte	https://www.youtube.com/watch?v=1CgrUZgLQlc&list=PLm8dSOaqLPHI65eHiBKtNrbgV47hGbOaM&index=16
20–23	Define and call user-defined functions. Demonstrate call by value and call by reference. Develop modular programs.	CLO04	Lecture	Test questions	https://www.geeksforgeeks.org/functions-in-c/ https://www.javatpoint	https://www.youtube.com/watch?v=7kxaZd4DZAU&list=PLm8dSOaqLPHI65eHiBKtNr

					.com/c-functions https://www.programiz.com/c-programming/c-functions https://en.cppreference.com/w/c/language/function	bgV47hGbOaM&index=13
24–25	Write recursive programs (factorial, Fibonacci, GCD). Demonstrate storage classes with suitable examples.	CLO04	Demonstration method using simulation tool	Test questions, Brainstorming	https://www.geeksforgeeks.org/recursion-in-c/ https://www.programiz.com/c-programming/c-recursion https://www.javatpoint.com/storage-classes-in-c	https://www.youtube.com/watch?v=ngCos392W4w https://www.youtube.com/watch?v=cAnujKR4stQ&list=PLm8dSOaqLPHI65eHiBKtNrbgV47hGbOaM&index=53
26–27	Practice declaring, initializing, and dereferencing pointers. Understand address manipulation using pointers.	CLO05	Lecture	Quiz, Test questions	https://www.includehelp.com/c/default.aspx?utm_source https://www.programiz.com/c-programming/c-pointer-examples https://www.learn-c.org/Pointers	https://www.youtube.com/watch?v=dc8euF3aIVc&list=PLm8dSOaqLPHI65eHiBKtNrbgV47hGbOaM&index=27
28	Demonstrate pointer arithmetic, pointers with arrays, and arrays of pointers using practical programs.	CLO05	Discussion	Test questions	https://www.includehelp.com/c/default.aspx?utm_source https://www.learn-c.org/Pointers	https://www.youtube.com/watch?v=pBrCD9A_i_w&list=PLm8dSOaqLPHI65eHiBKtNrbgV47hGbOaM&index=28
29–30	Pass pointers to functions, return pointers, and introduce function pointers.	CLO04	Demonstration method using simulation tool	Quiz, Group discussion	https://www.includehelp.com/c/default.aspx?utm_source https://www.learn-c.org/Pointers	https://www.youtube.com/watch?v=dc8euF3aIVc&list=PLm8dSOaqLPHI65eHiBKtNrbgV47hGbOaM&index=27
31–32	Implement dynamic memory allocation using malloc(), calloc(), realloc(), and free(). Identify memory leaks.	CLO05	Lecture	Test questions	https://www.includehelp.com/c/default.aspx?utm_source https://www.learn-c.org/en/Dynamic_allocation	https://www.youtube.com/watch?v=LQXueqsbxRU
33–34	Define and use structures. Create arrays of structures and access members using structure pointers.	CLO06	Discussion	Test questions	https://www.includehelp.com/c/default.aspx?utm_source	https://www.youtube.com/watch?v=T6eSt06zuaA&list=PLm8dSOaqLPHI65eHiBKtNrbgV47hGbOaM&index=11 https://www.youtube.com/watch?v=KChOkB7aTS4&list=PLBlnK6fEyqRiteqwlMLXYtZ16xXDR7MO0&index=11
35–36	Implement unions and compare memory utilization with structures through example programs.	CLO06	Lecture	Test questions	https://www.includehelp.com/c/default.aspx?utm_source	https://www.youtube.com/watch?v=CjROf4vRwFk&list=PLm8dSOaqLPHI65eHiBKtNrbgV47hGbOaM&index=20 https://www.youtube.com/watch?v=ECNDwnfQXyQ&list=PLm8dSOaqLPHI65eHiBKtNr

						bgV47hGbOaM&index=21
37	Use enumerations and typedef to create readable and efficient programs.	CLO06	Discussion	Quiz	https://www.includehelp.com/c/default.aspx?utm_source	https://www.youtube.com/watch?v=CjROf4vRwFk&list=PLm8dSOaqLPHI65eHiBKtNrbgV47hGbOaM&index=20 https://www.youtube.com/watch?v=9QdJExC2AVg&list=PLBlnK6fEyqRiteqwlMLXYtZ16xXDR7MO0&index=21
38–39	Perform file operations: open, close, read, and write files using various file modes.	CLO06	Demonstration method using simulation tool	Test questions	https://www.includehelp.com/c-programs/file-handling-in-c-solved-programs.aspx https://www.programiz.com/c-programming/c-file-examples	https://www.youtube.com/watch?v=4OGMB4Fhh50&list=PLBlnK6fEyqRhX6r2uhhlubuF5QextdCSM
40–41	Use file I/O functions and implement random file access using fseek(), ftell(), and rewind().	CLO06	Lecture	Test questions	https://www.includehelp.com/c-programs/file-handling-in-c-solved-programs.aspx	https://www.youtube.com/watch?v=4OGMB4Fhh50&list=PLBlnK6fEyqRhX6r2uhhlubuF5QextdCSM
42–43	Apply preprocessor directives and implement programs using command-line arguments (argc, argv).	CLO06	Discussion	Quiz	https://www.tutorialspoint.com/cprogramming/c_working_of_preprocessor.htm	https://www.youtube.com/watch?v=QHZYhkI-V34 https://www.youtube.com/watch?v=pRMZCETp71g https://www.youtube.com/watch?v=YZTgWvJvZrM
44–45	Implement bit fields and variable argument functions using <stdarg.h> with practical system-level examples.	CLO06	Lecture	Test questions	https://www.tutorialspoint.com/cprogramming/c_command_line_arguments.htm	https://www.youtube.com/watch?v=QHZYhkI-V34

8. Remedial Classes

After every sessional test, different types of learners will be identified, and special discussions will be planned and scheduled accordingly.

Action Plan for different types of learners:

Learner Type-I	Learner Type- II	Learner Type- III
Remedial Classes, Doubt Sessions, Guided Tutorials	Workshop, Doubt Session	Coding Competitions, Project

9. Self-Learning

Assignments to promote self-learning, survey of contents from multiple sources.

S.No	Topics	CLO	ALM[8]	References/MOOCS
1.	Data types in C, Basic input/ output functions.	CLO01, CLO02	Think – Pair- Share, Peer Review	https://nptel.ac.in/courses/106106210 https://www.coursera.org/learn/c-for-everyone

				https://ocw.mit.edu/courses/6-087-practical-programming-in-c-january-iap-2010/pages/calendar/
2.	Conditional Statements, Iterations and Functions	CLO03, CLO04	Think – Pair- Share, Brain Storming sessions	https://nptel.ac.in/courses/106106210 https://www.coursera.org/learn/c-for-everyone

10. Delivery Details of Content Beyond Syllabus

Content beyond syllabus covered (if any) should be delivered to all students that would be planned, and schedule notified accordingly.

S.No	Advanced Topics, Additional Reading, Research papers and any	CLO	POs	ALM[8]	References/MOOCS
1	Learn file handling in C. Focus on opening, reading from, writing to, and closing files.	CLO06	PO1, PO3, PO4, PO5, PO11	Think – Pair- Share, Peer Review	https://nptel.ac.in/courses/106104128

11. Evaluation Scheme & Components:

Assessment Type[12]	Evaluation Component[9]	Type of Component[10]	No. of Assessments[11]	% Weightage of Component	Max. Marks	Mode of Assessment	CLO
Formative	Component 1	Continuous Lab Evaluations	01*	20%	20	Viva Voce, Practical, and File	CLO01-CLO06
Summative	Component 2	Sessional Tests (STs)	02**	30%	30	Computer Based Test (CBT)	CLO01-CLO06
Summative	Component 3	End Term Examination	01***	50%	50	Computer Based Test (CBT)	CLO01-CLO06
	Total		**100%**				

*Note: *Continuous Evaluation (CE) is a mandatory evaluation taken once in a semester.*

***All STs are mandatory.*

****As per academic guidelines, a minimum of 75% attendance is required to appear in the end-of-semester examination.*

.

12. Syllabus of the Course:

Subject: Fundamentals of C Programming			
S.No.	**Topic (s)**	**No. of Sessions**	**Weightage %**
1	History of C, understanding its role in system programming and as a foundation for other languages, Preprocessor directives, the main() function, comments, and basic syntax. Input and Output- Using printf() for output and scanf() for input Variables and Constants- Declaration, initialization, and naming conventions. Data Types-Basic types: int, char, float, double, Modifiers: signed, unsigned, short, long. Operators-Arithmetic, relational, and logical operators, Assignment, increment/decrement, and bitwise operators Ternary operator, Type Conversion- Implicit and explicit casting Statements- if, if-else, else-if ladder, and switch-case Loops (Iteration): for, while, and do-while loops; Jump Statements—break, continue, and goto Arrays- Declaration, initialization, and accessing elements of one-dimensional arrays, Multidimensional arrays (2D arrays).	**19**	**42%**

	Strings: Introduction to strings as character arrays, String manipulation using standard library functions (strcpy, strcat, strlen, strcmp, etc.) from <string.h>.		
ST-1 (Covering 49% syllabus, Lect. No. 1-19)			
2	Functions and Recursion: Definition, declaration (prototype), and calling functions Types of Functions- User-defined vs. standard library functions, Parameter Passing- Call by value vs. call by reference. Recursion- Understanding recursive functions and their applications (e.g., factorial, Fibonacci series). Storage Classes- auto, extern, static, and register. Pointers: Understanding memory addresses, declaration, and initialization of pointers, Pointer Operators- Address-of operator (&) and dereference operator (*). Pointer Arithmetic- Incrementing and decrementing pointers, Relationship between arrays and pointers, pointer to an array, and array of pointers, Passing pointers to functions and returning pointers from functions, Function Pointers: Pointers that point to functions. Dynamic Memory Allocation: Memory Management Functions-using malloc(), calloc(), realloc(), and free() from <stdlib.h>, Memory Leaks: Understanding and avoiding memory leaks User-Defined Data Types: Structures- Defining, declaring, and accessing structure members. Arrays of structures and pointers to structures Unions- Defining and using unions, understanding the memory difference between structures and unions Enumerations- Creating and using enumerated data types for named integer constants. typedef- Creating aliases for existing data types.	**18**	**40%**
ST-2 (Covering 42% syllabus, Lect. No. 20-37)			
3	File Operations: Opening, closing, reading, and writing files, File Modes- Understanding different file access modes (r, w, a, r+, w+, a+) File I/O Functions-fopen(), fclose(), fprintf(), fscanf(), fgetc(), fputc(), fgets(), fputs(), fread(), fwrite(), Random Access to Files- Using fseek(), ftell(), and rewind(). Preprocessor Directives- #include, #define, #if, #else, #endif, macros, Command Line Arguments- Passing arguments to the main() function (argc and argv) Bit Fields- Using bit fields inside structures to save memory.Variable Argument Lists- Using <stdarg.h> to create functions that accept a variable number of arguments.	**08**	**18%**
End Term (Covering (42%+40%+18%)=100% syllabus)			

13. Academic Integrity Policy:

Education at Chitkara University builds on the principle that excellence requires freedom where Honesty and integrity are its prerequisites. Academic honesty in the advancement of knowledge requires that all students and Faculty respect the integrity of one another' s work and recognize the importance of acknowledging and safeguarding intellectual property. Any breach of the same will be tantamount to severe academic penalties.

This Document is approved by:

Designation	Name	Signature
Course Coordinator	Dr. Gaganpreet Kaur	
Program Incharge	Dr. Preetinder Singh Brar	
Pro Vice Chancellor	Dr. Jaiteg Singh	
Date(DD/MM/YYYY)	28.1.26	