

AN ISO 9001:2015 CERTIFIED COMPANY



DREAM INDIA
= TECHNOLOGIES =

C Programming

SYLLABUS

DREAM INDIA TECHNOLOGIES

CALL - 9985943539, 9675596999

E-MAIL: info@dreamindiatechnologies.com | HYDERABAD | VIJAYAWADA | GUNTUR



C Programming

C

SYLLABUS

COVERAGE

- | | |
|---|---|
| <input type="checkbox"/> Introduction to Programming | <input type="checkbox"/> Dynamic memory allocation |
| <input type="checkbox"/> Fundamentals in C | <input type="checkbox"/> String |
| <input type="checkbox"/> Operators and Expressions | <input type="checkbox"/> String and function |
| <input type="checkbox"/> Data types | <input type="checkbox"/> Command line arguments |
| <input type="checkbox"/> Input-Output Library Functions | <input type="checkbox"/> Preprocessor |
| <input type="checkbox"/> Control statements | <input type="checkbox"/> Structure |
| <input type="checkbox"/> Function | <input type="checkbox"/> Structure and function |
| <input type="checkbox"/> Storage class | <input type="checkbox"/> File Handling |
| <input type="checkbox"/> Pointer | <input type="checkbox"/> ODBC Programming |
| <input type="checkbox"/> Pointer and Function | <input type="checkbox"/> Process and threads |
| <input type="checkbox"/> Array | <input type="checkbox"/> Graphics |
| <input type="checkbox"/> Pointer and array | <input type="checkbox"/> Socket and Network programming |
| <input type="checkbox"/> Array and function | <input type="checkbox"/> Project |

SYLLABUS IN DETAILS

Introduction to Programming

- ☐ Program and Programming
- ☐ Programming Languages
- ☐ Types of software's
- ☐ Operating Systems
- ☐ Dos commands
- ☐ Basic Linux commands and vi editor
- ☐ Compiler, Interpreter, Loader and Linker

Fundamentals in C

- ☐ History of 'C'
- ☐ A Simple C Program
- ☐ Program execution phases
- ☐ Backslash character constants
- ☐ Character set
- ☐ Constants
- ☐ Number systems
- ☐ Format specifiers

- ☐ Identifiers ☐ Keywords
- ☐ Variables
- ☐ Data Types
- ☐ Declaration of Variable
- ☐ Assigning Values to Variables
- ☐ Initialization
- ☐ Comments
- ☐ Const Qualifier
- ☐ Basic Structure of a 'C' program
- ☐ Programming Examples

Operators and Expressions

- ☐ Dealing with all 45 operators
- ☐ Arithmetic operators
- ☐ Increment and decrement operators
- ☐ Relational operators
- ☐ Logical operators
- ☐ The bitwise operators

- ☐ The assignment operators
- ☐ The conditional operator
- ☐ The size of operator
- ☐ The comma operator
- ☐ Type casting operator
- ☐ Other operators
- ☐ Precedence and order of evaluation
- ☐ Programming Examples
- ☐ FAQ's

Data types

- ☐ Modifiers
- ☐ Format specifiers
- ☐ Dealing with each data types
- ☐ Memory representation of each type
- ☐ Programming Examples

Input-Output Library Functions

- ☐ Unformatted I-O Functions
- ☐ Single Character Input-Output
- ☐ String Input-Output
- ☐ Formatted I-O Functions
- ☐ printf() Width Specifier
- ☐ scanf() Width Specifier
- ☐ Programming Examples

Control statements

- ☐ Conditional Control Statements
- ☐ if
- ☐ if-else
- ☐ nested if-else
- ☐ else-if ladder
- ☐ Multiple Branching Control Statement
- ☐ switch-case
- ☐ Loop Control Statements
- ☐ while
- ☐ do-while
- ☐ for
- ☐ Nested Loops
- ☐ Jump Control statements
- ☐ break
- ☐ continue
- ☐ goto
- ☐ exit
- ☐ return

- ☐ Programming Examples
- ☐ FAQ's

Function

- ☐ What is function?
- ☐ Why function?
- ☐ Advantages of using functions
- ☐ Function Prototype
- ☐ Defining a function
- ☐ Calling a function
- ☐ Return statement
- ☐ Types of functions
- ☐ Recursion
- ☐ Nested functions
- ☐ main() function
- ☐ Library Function
- ☐ Local and global variables
- ☐ Programming Examples
- ☐ FAQ's

Storage class

- ☐ Types of storage class
- ☐ Scoping rules
- ☐ Dealing with all storage classes
- ☐ Programming Examples
- ☐ FAQ's

Pointer

- ☐ Def of Pointer
- ☐ Declaration of Pointer Variables
- ☐ Assigning Address to Pointer Variables
- ☐ De-referencing Pointer Variables
- ☐ Pointer to Pointer
- ☐ Pointer Arithmetic
- ☐ Pointer comparisons
- ☐ De-reference and increment pointer
- ☐ pointer to const data
- ☐ const pointer
- ☐ const pointer to const data
- ☐ Void pointer or Generic Pointer
- ☐ Null pointer
- ☐ wild pointer
- ☐ Programming Examples
- ☐ FAQ's

Pointer and Function

- ☐ Parameter Passing Techniques – call by value, call by address
- ☐ Using Pointers as Arguments
- ☐ Function Returning value
- ☐ Returning More than one value From A Function
- ☐ Functions Returning Address
- ☐ Function Returning Pointers
- ☐ Dangling pointer
- ☐ Pointer to a Function
- ☐ Calling A function through function pointer
- ☐ passing A function's address as an Argument to other function
- ☐ Functions with variable number of arguments
- ☐ Programming Examples
- ☐ FAQ's

Array

- ☐ One dimensional arrays
- ☐ Declaration of 1D arrays
- ☐ Initialization of 1D arrays
- ☐ Accessing element of 1D arrays
- ☐ Reading and displaying elements
- ☐ Two dimensional arrays
- ☐ Declaration of 2D arrays
- ☐ Initialization of 2D arrays
- ☐ Accessing element of 2D arrays
- ☐ Reading and displaying elements
- ☐ Programming Examples

FAQ's Pointer and Array

- ☐ Pointer and one dimensional arrays
- ☐ Subscripting pointer variables
- ☐ Pointer to an array
- ☐ Array of pointers
- ☐ Pointers and two dimensional arrays
- ☐ Subscripting pointer To an array
- ☐ Programming Examples
- ☐ FAQ's

Array and Function

- ☐ 1D array and function
- ☐ Passing individual array elements to a function
- ☐ passing individual array elements address to a function
- ☐ passing whole 1d array to a function
- ☐ 2D array and function
- ☐ Passing individual array elements to a function
- ☐ passing individual array elements address to a function
- ☐ passing whole 2d array to a function
- ☐ using arrays of function pointer
- ☐ Programming Examples
- ☐ FAQ's

Dynamic memory allocation

- ☐ malloc()
- ☐ calloc()
- ☐ realloc()
- ☐ free()
- ☐ Core dump
- ☐ Memory leak
- ☐ Dynamic 1D and 2D Arrays
- ☐ Programming Examples
- ☐ FAQ's

Strings

- ☐ strings versus character arrays
- ☐ Initializing strings
- ☐ Reading string
- ☐ Displaying string
- ☐ The %s format specifier
- ☐ The gets() and puts() functions
- ☐ string handling functions
- ☐ string pointers
- ☐ Two-dimensional character arrays or array of string
- ☐ array of pointers to strings
- ☐ Programming Examples
- ☐ FAQ's

Command line arguments

- ☐ what is command prompt?
- ☐ why command line?
- ☐ What are command line arguments?
- ☐ Programs using command line

Preprocessor

- ☐ What is preprocessing?
- ☐ Macro expansions
- ☐ File inclusions
- ☐ Conditional compilation
- ☐ The stringification(#)and token passing operator
- ☐ (##) operators
- ☐ Programming Examples
- ☐ FAQ's

Structure

- ☐ Why is structure used?
- ☐ What is structure?
- ☐ Advantages of structures
- ☐ Defining a Structure
- ☐ Declaration of Structure Variables
- ☐ Initialization of Structure Variables
- ☐ Accessing Structure Members
- ☐ Storage of Structures in Memory
- ☐ Size of Structures
- ☐ Reading and Displaying Structure Variables
- ☐ Assignment of Structure Variables
- ☐ Pointers to structures
- ☐ Array of structures
- ☐ Arrays within structures
- ☐ Nested structures
- ☐ Self-referential structures
- ☐ memory link(linked list)
- ☐ Bit fields
- ☐ Programming Examples
- ☐ FAQ's

Structure and Function

- ☐ Passing structure member to a function
- ☐ Passing structure variable to a function

☐ Passing structure variable address to a function

- ☐ Passing array of structure to a function
- ☐ Returning a structure variable from function
- ☐ Returning a structure variable address from function
- ☐ Returning structure variable from a function
- ☐ Programming Examples
- ☐ FAQ's

Union and Enumeration and typedef

- ☐ What are unions?
- ☐ Structures versus unions
- ☐ Working with unions
- ☐ Initializing unions
- ☐ Advantages of unions
- ☐ enum keyword
- ☐ typedef keyword
- ☐ Programming Examples
- ☐ FAQ's

File Handling

- ☐ Using files in C
- ☐ Buffer and streams
- ☐ Working with text files and Binary Files
- ☐ File operations using std. library and system calls
- ☐ File management I/O functions
- ☐ Random Access Files
- ☐ Programming Examples
- ☐ FAQ's

ODBC Programming

- ☐ ODBC rules and regulation
- ☐ Introduction to MYSQL and Oracle
- ☐ Creating, inserting and retrieving records for different Data bases.
- ☐ Programming Examples
- ☐ FAQ's



Process and Threads

- ☐ What is process & Threads
- ☐ Use of fork, vfork
- ☐ Daemon process
- ☐ Programming Examples
- ☐ FAQ's

Graphics & Curses

- ☐ Graphics using Glade interface with GTK+
- ☐ Working with GTK Widgets, Event handling
- ☐ Developing Application Interface
- ☐ Programming Examples
- ☐ FAQ's

Socket and Network programming

Project

AN ISO 9001:2015 CERTIFIED COMPANY



DREAM INDIA
= TECHNOLOGIES =