**Dreamer Infotech**

**C Programming Syllabus**

**1. C Basics**

**1.1. C Language Introduction**

**1.2. Setting Up C Development Environment**

**1.3. C Hello World Program**

**1.4. Compiling a C Program: Behind the Scenes**

**1.5. Tokens in C**

**1.6. C Identifiers**

**1.7. Keywords in C**

**---**

**2. C Variables and Constants**

**2.1. C Variables and Constants**

**2.2. C Variables**

**2.3. Constants in C**

**2.4. Const Qualifier in C**

**2.5. Different Ways to Declare Variable as Constant in C**

**2.6. Scope Rules in C**

**2.7. Internal Linkage and External Linkage in C**

**2.8. Global Variables in C**

**---**

**3. C Data Types**

**3.1. Data Types in C**

**3.2. Data Type Modifiers in C**

**3.3. Literals in C**

**3.4. Escape Sequence in C**

**3.5. bool in C**

**3.6. Character Arithmetic in C**

**3.8. Type Conversion in C**

**---**

**4. C Input/Output**

**4.1. Basic Input and Output in C**

**4.2. Format Specifiers in C**

**4.3. printf in C**

**4.4. scanf in C**

**4.5. Scansets in C**

**4.6. Formatted and Unformatted Input and Output Functions**

**---**

**5. C Operators**

**5.1. Operators in C**

**5.2. Arithmetic Operators in C**

**5.3. Unary Operators in C**

**5.4. Relational Operators in C**

**5.5. Bitwise Operators in C**

**5.6. Logical Operators in C**

**5.7. Assignment Operators in C**

**5.8. Increment and Decrement Operators in C**

**5.9. Conditional or Ternary Operator (?:) in C**

**5.10. size of Operator in C**

**5.11. Operator Precedence and Associativity in C**

**---**

**6. C Control Statements**

**6.1. Decision-Making in C**

**6.2. C if Statement**

**6.3. C if…else Statement**

**6.4. C if-else-if Ladder**

**6.5. Switch Statement in C**

**6.6. Using Range in switch case in C**

**---**

**7. Loops in C**

**7.1. C for Loop**

**7.2. while Loop in C**

**7.3. do…while Loop in C**

**7.4. for versus while Loop**

**7.5. continue Statement in C**

**7.6. break Statement in C**

**7.7. goto Statement in C**

**---**

**8. C Functions**

**8.1. C Functions**

**8.2. User-Defined Function in C**

**8.3. Parameter Passing Techniques in C**

**8.4. Importance of Function Prototype in C**

**8.5. Return Multiple Values From a Function**

**8.6. main Function in C**

**8.7. Implicit Return Type int in C**

**8.8. Callbacks in C**

**8.9. Nested Functions in C**

**8.10. Variadic Functions in C**

**8.11. Maths Functions in C**

**---**

**9. C Arrays & Strings**

**9.1. C Arrays**

**9.2. Properties of Array in C**

**9.3. Multidimensional Arrays in C**

**9.4. Initialization of Multidimensional Arrays in C**

**9.5. Pass Array to Functions in C**

**9.6. Pass a 2D Array as a Parameter in C**

**9.7. Data Types for Which Array is Not Possible**

**9.8. Pass an Array by Value in C**

**9.9. Strings in C**

**9.10. An Array of Strings in C**

**9.11. Difference Between Single Quoted and Double Quoted Initialization**

**9.12. String Functions in C**

**---**

**10. C Pointers**

**10.1. C Pointers**

**10.2. Pointer Arithmetics in C**

**10.3. Pointer to Pointer (Double Pointer) in C**

**10.4. Function Pointer in C**

**10.5. Declare Function Pointer in C**

**10.6. Pointer to an Array in C**

**10.7. Constant Pointer in C**

**10.8. Pointer vs Array in C**

**10.9. Dangling, Void, Null and Wild Pointers**

**10.10. Near, Far and Huge Pointers in C**

**10.11. restrict Keyword in C**

**---**

**11. C User-Defined Data Types**

**11.1. C Structures**

**11.2. dot (.) Operator in C**

**11.3. C typedef**

**11.4. Structure Member Alignment, Padding and Data Packing**

**11.5. Flexible Array Members in a Structure in C**

**11.6. C Unions**

**11.7. Bit Fields in C**

**11.8. Difference Between Structure and Union in C**

**11.9. Anonymous Union and Structure in C**

**11.10. Enumeration (or enum) in C**

**---**

**12. C Storage Classes**

**12.1. Storage Classes in C**

**12.2. extern Keyword in C**

**12.3. Static Variables in C**

**12.4. Initialization of Static Variables in C**

**12.5. Static Functions in C**

**12.6. Understanding “volatile” Qualifier in C**

**12.7. Understanding the “register” Keyword in C**

**---**

**13. C Memory Management**

**13.1. Memory Layout of C Programs**

**13.2. Dynamic Memory Allocation in C**

**13.3. Difference Between malloc() and calloc()**

**13.4. What is a Memory Leak?**

**13.5. Dynamic Array in C**

**13.6. Dynamically Allocate a 2D Array in C**

**13.7. Dynamically Growing Array in C**

**---**

**14. C Preprocessor**

**14.1. C Preprocessors**

**14.2. C Preprocessor Directives**

**14.3. How a Preprocessor Works in C?**

**14.4. Header Files in C**

**14.5. Difference Between Header Files “stdio.h” and “stdlib.h”**

**14.6. Write Your Own Header File in C**

**14.7. Macros and their Types in C**

**14.8. Interesting Facts About Macros and Preprocessors in C**

**14.9. and Operators in C**

**14.10. Print a Variable Name in C**

**14.11. Multiline Macros in C**

**14.12. Variable Length Arguments for Macros**

**14.13. Branch Prediction Macros in GCC**

**14.14. typedef versus define in C**

**14.15. Difference Between define and const in C**

**---**

**15. C File Handling**

**15.1. Basics of File Handling in C**

**15.2. C fopen() Function**

**15.3. EOF, getc() and feof() in C**

**15.4. fgets() and gets() in C**

**15.5. fseek() vs rewind() in C**

**15.6. Return Type of getchar(), fgetc() and getc()**

**15.7. Read/Write Structure From/to a File in C**

**15.8. C Program to Print Contents of File**

**15.9. C Program to Delete a File**

**15.10. C Program to Merge Contents of Two Files into a Third File**

**15.11. Difference Between printf, sprintf and fprintf**

**15.12. Difference Between getc(), getchar(), getch() and getche()**

**---**

**16. C Error Handling**

**16.1. Error Handling in C**

**16.2. Using goto for Exception Handling in C**

**16.3. Error Handling During File Operations in C**

**16.4. C Program to Handle Divide By Zero and Multiple Exceptions**

**---**

**17. C Programs**

**17.1. Basic C Programs**

**17.2. Control Flow Programs**

**17.3. Pattern Printing Programs**

**17.4. Functions Programs**

**17.5. Arrays Programs**

**17.6. Strings Programs**

**17.7. Conversions Programs**

**17.8. Pointers Programs**

**17.9. Structures and Unions Programs**

**17.10. File I/O Programs**

**17.11. Date and Time Programs**

**---**

**C++ Programming Syllabus**

**---**

**1. C++ Input/Output**

**1.1. C++ Basic Input / Output**

**1.2. C++ Standard Input Stream (cin)**

**1.3. C++ Standard Output Stream (cout)**

**1.4. C++ Standard Error Stream (cerr)**

**1.5. C++ Input / Output Manipulator**

**---**

**2. C++ Functions**

**2.1. C++ Functions**

**2.2. C++ return**

**2.3. C++ Parameter Passing Techniques**

**2.4. Difference Between Call by Value and Call by Reference**

**2.5. C++ Default Arguments**

**2.6. C++ Recursion**

**2.7. C++ Inline Functions**

**2.8. C++ Lambda Expression**

**---**

**3. C++ Strings**

**3.1. C++ Strings**

**3.2. C++ std::string Class**

**3.3. C++ Array of Strings**

**3.4. C++ String Functions**

**3.5. C++ String Concatenation**

**3.6. Tokenizing a String in C++**

**3.7. C++ Substring**

**---**

**4. C++ Structures and Unions**

**4.1. C++ Structures, Unions, and Enumerations**

**4.2. C++ Structures**

**4.3. C++ Pointer to Structure**

**4.4. C++ Self-Referential Structures**

**4.5. Difference Between C Structures and C++ Structures**

**4.6. C++ Unions**

**4.7. C++ Bit Fields**

**4.8. C++ Enumeration**

**4.9. C++ typedef**

**4.10. Array of Structures vs Array within a Structure in C/C++**

**---**

**5. C++ Dynamic Memory Management**

**5.1. C++ Dynamic Memory Management**

**5.2. C++ new and delete Operators**

**5.3. new vs malloc() and free() vs delete in C++**

**5.4. Memory Leak in C++**

**5.5. Difference Between Static and Dynamic Memory Allocation in C++**

**---**

**6. C++ Object-Oriented Programming**

**6.1. C++ Object Oriented Programming (OOPs)**

**6.2. C++ Classes and Objects**

**6.3. C++ Access Modifiers**

**6.4. C++ Friend Class and Function**

**6.5. C++ Constructors**

**6.6. C++ Default Constructors**

**6.7. C++ Copy Constructor**

**6.8. C++ Destructors**

**6.9. C++ Private Destructor**

**6.10. When is the Copy Constructor Called?**

**6.11. Shallow Copy and Deep Copy in C++**

**6.12. When Should We Write Our Own Copy Constructor?**

**6.13. Does the Compiler Create a Default Constructor When We Write Our Own?**

**6.14. C++ Static Data Members**

**6.15. C++ Static Member Functions**

**6.16. C++ this Pointer**

**6.17. C++ Scope Resolution Operator vs this Pointer**

**6.18. C++ Local Class**

**6.19. C++ Nested Classes**

**6.20. C++ enum Class**

**6.21. Difference Between Structure and Class in C++**

**---**

**7. C++ Encapsulation and Abstraction**

**7.1. C++ Encapsulation**

**7.2. C++ Abstraction**

**7.3. Difference Between Abstraction and Encapsulation in C++**

**---**

**8. C++ Polymorphism**

**8.1. C++ Polymorphism**

**8.2. C++ Function Overriding**

**8.3. C++ Virtual Functions and Runtime Polymorphism**

**8.4. Difference Between Compile-time and Run-time Polymorphism in C++**

**8.5. Difference Between Inheritance and Polymorphism in C++**

**---**

**9. C++ Function Overloading**

**9.1. C++ Function Overloading**

**9.2. C++ Constructor Overloading**

**9.3. Functions that Cannot be Overloaded**

**9.4. Function Overloading and const Keyword**

**9.5. Function Overloading and Return Type**

**9.6. Function Overloading and float Data Type**

**9.7. Function Overloading and Default Arguments**

**9.8. Can main() be Overloaded?**

**9.9. Function Overloading vs Function Overriding**

**9.10. Advantages and Disadvantages of Function Overloading**

**---**

**10. C++ Operator Overloading**

**10.1. C++ Operator Overloading**

**10.2. Types of Operator Overloading**

**10.3. C++ Functors**

**10.4. Operators that Cannot be Overloaded**

**---**

**11. C++ Inheritance**

**11.1. C++ Inheritance**

**11.2. C++ Inheritance Access**

**11.3. C++ Multiple Inheritance**

**11.4. C++ Hierarchical Inheritance**

**11.5. C++ Multilevel Inheritance**

**11.6. Constructor in Multiple Inheritance**

**11.7. Inheritance and Friendship**

**11.8. Does Function Overloading Work with Inheritance?**

**11.9. Difference Between Inheritance and Polymorphism**

**---**

**13. C++ Files and Streams**

**13.1. C++ Files and Streams**

**13.2. C++ I/O Redirection**

**---**

**14. C++ Templates**

**14.1. C++ Templates**

**14.2. C++ Template Specialization**

**14.3. C++ using Keyword**

**---**

**15. C++ Standard Template Library (STL)**

**15.1. The C++ Standard Template Library (STL)**

**15.2. STL Algorithms**

**15.3. STL Containers**

**15.4. STL Vector**

**15.5. STL Pair**

**15.6. STL Set**

**15.7. STL Multiset**

**15.8. STL Stack**

**15.9. STL Queue**

**15.10. STL Priority Queue**

**15.11. STL Deque**

**15.12. STL List**

**15.13. STL Forward List**

**15.14. STL Map**

**15.15. STL Multimap**

**15.16. STL Bitset**

**15.17. STL Unordered Sets**

**15.18. STL Unordered Multiset**

**15.19. STL Unordered Map**

**15.20. STL Unordered Multimap**

**---**

**16. C++ Iterators**

**16.1. Introduction to C++ Iterators**

**16.2. C++ Input Iterators**

**16.3. C++ Output Iterators**

**16.4. C++ Forward Iterators**

**16.5. C++ Bidirectional Iterators**

**16.6. C++ Random Access Iterators**

**16.7. C++ istream\_iterator and ostream Iterator**

**16.8. Difference Between C++ Iterators and Pointers**

**---**

**17. Advanced C++**

**17.1. C++ Multithreading**

**17.2. C++ Smart Pointers**

**17.3. Differences Between Different C++ Smart Pointers**

**17.4. Type of ‘this’ Pointer in C++**

**17.5. Delete ‘this’ Pointer in C++**

**17.6. Passing C++ Function as a Parameter**

**17.7. C++ Signal Handling**

**17.8. C++ Generics**

**---**