Mindluster Platform

About the course Symbolic Constants In C++ Programming

Symbolic Constants In C++ course in hindi, in this course we will learn about the Symbolic Constants in C++, essential for defining fixed values that do not change during program execution. Symbolic constants enhance code readability, maintainability, and reduce errors. We will explore two primary methods for defining symbolic constants: using the #define preprocessor directive and the const keyword. The #define directive allows for creating macro constants, which are replaced by their values during preprocessing. The const keyword, on the other hand, defines constants with type safety and scope control. We'll delve into the differences between #define and const, highlighting the advantages and use cases of each. Additionally, we will discuss the scope and lifetime of symbolic constants, ensuring you understand where and how long these constants exist in memory. The course will cover best practices for naming constants, emphasizing readability and maintainability. Furthermore, we will explore the role of constants in object-oriented programming, including how to define constants within classes and objects. You'll learn about the benefits of using symbolic constants in functions and arrays, as well as the use of enumerated constants (enum) for creating a set of named integer constants. Through practical examples and hands-on exercises, you will gain experience in defining and using symbolic constants effectively in your C++ programs. By the end of this course, you will have a comprehensive understanding of symbolic constants, enabling you to write cleaner, more efficient, and error-free code. Join us to master the use of symbolic constants in C++ and elevate your programming skills.

Programming Category's Courses

Course Lesson(83)

Lesson 1 : **History of C Origins of c Evolution of c introduction to c programming**

Lesson 2 : **Basic Concepts Principles of OOP in C Object Oriented Programming CPP**

Lesson 3 : **C vs C Differences between C and CPP versus**

Lesson 4 : **Basic Structure of C Program C Programming**

Lesson 5 : **cout and cin objects In C Input Stream Output Stream Input Operator Output Operator**

Lesson 6 : **A Sample C Program c program to add two numbers cout object in c**

Lesson 7 : **Dynamic Initialization of Variables In C Programming**

Lesson 8 : **new and delete operators in C Memory Management Operators In C Programming**

Lesson 9 : **Keywords In C Programming**

Lesson 10 : **Identifiers In C ProgrammingÂ Variables In C Language**

Lesson 11 : **Constants In C ProgrammingÂ Literals In C Language**

Lesson 12 : **Symbolic Constants In C ProgrammingÂ Defining Constants In C Language**

Lesson 13 : **Data Types In C Programming**

Lesson 14 : **Operators In C Programming**

Lesson 15 : **Scope Resolution Operator In C Programming**

Lesson 16 : **Manipulators In C Programming**

Lesson 17 :

SIMPLE IF IN C IF IN CPP CONDITIONAL DECISION BRANCHING CONTROL STATEMENTS IN CPP

Lesson 18 :
IF ELSE IN C CPP CONDITIONAL DECISION BRANCHING CONTROL STATEMENTS IN CPP

Lesson 19 :
ELSE IF LADDER IN C CONDITIONAL DECISION BRANCHING CONTROL STATEMENTS IN CPP

Lesson 20 : **NESTED IF ELSE In C CPP Conditional Branching Decision Making Control Statements**

Lesson 21 : **Switch in C Selection control statement in CPP**

Lesson 22 : **FOR LOOP IN C LOOPING CONTROL STATEMENTS IN CPP**

Lesson 23 : **WHILE LOOP IN C LOOPING CONTROL STATEMENTS IN CPP**

Lesson 24 : **DO WHILE LOOP IN C LOOPING CONTROL STATEMENTS IN CPP**

Lesson 25 : **Arrays in C One dimensional Array in CPP**

Lesson 26 : **Multi Dimensional Arrays in C Two Dimensional Arrays in CPP Matrix in cpp**

Lesson 27 :
FUNCTIONS IN C definition declaration call actual and formal arguments calling called function

Lesson 28 : **Categories Types of Function Definitions in C Function Design strategies in CPP**

Lesson 29 : **Recursion in C Factorial of a given number using Recursion in CPP**

Lesson 30 : **Storage Classes in C**

Lesson 31 : **Structures in C**

Lesson 32 : **Class and Objects in C Accessing Class Members Class Declaration in C Object Creation**

Lesson 33 : **Defining Member Functions in C Inside the Class Outside the Class C Programming**

Lesson 34 : **Static Data Members in C Static Member Variables in c C programming**

Lesson 35 : **Static Member Functions in C C Programming**

Lesson 36 : **Friend Function In C Programming**

Lesson 37 :
Friend Function Program C Program to add two numbers between 2 classes using friend function

Lesson 38 :
Friend Function Program C Program to swap two members between 2 classes using Friend Function

Lesson 39 : **Friend Class in C Programming**

Lesson 40 : **Inline Functions in C C Programming**

Lesson 41 : **Nested Member Function in C**

Lesson 42 : **Private Member Function in C**

Lesson 43 : **Nested Classes In C programming**

Lesson 44 : **Default Arguments In C Programming**

Lesson 45 : **Reference Variables in C CPP programming**

Lesson 46 :
Passing Objects as function arguments in C Programming Passing objects to functions in c

Lesson 47 : **Array Of Objects In C programming**

Lesson 48 : **Returning Objects In C Programming Object Assignment In C Language**

Lesson 49 :
Parameter Passing Techniques In C Passing Arguments to functions in cpp Pass by reference

Lesson 50 : **Arrays Inside Class in C**

Lesson 51 :
Constructors in C Programming Default Constructor Parameterized Constructor Copy Constructor

Lesson 52 : **Constructor Overloading In C**

Lesson 53 : **Dynamic Constructor in C CPP programming**

Lesson 54 : **Destructors In C Programming**

Lesson 55 : **Inheritance In C**

Lesson 56 : **Constructors in Derived Classes in C**

Lesson 57 : **Virtual Base Class In C Multipath Inheritance In C**

Lesson 58 : **POINTERS IN C Introduction to pointers in CPP**

Lesson 59 : **this pointer in C CPP**

Lesson 60 : **Pointer to Array in C**

Lesson 61 : **Pointer to object in C**

Lesson 62 : **Pointer to Base Class and Derived Class in C CPP**

Lesson 63 : **Function Overloading In C programming**

Lesson 64 : **Unary Operator Overloading In C using Member Function and Friend Function**

Lesson 65 : **Binary Operator Overloading In C using Member Function and Friend Function**

Lesson 66 : **Overloading Assignment Operator C Program**

Lesson 67 : **overloading insertion and extraction operator in c C Program**

Lesson 68 :
C program to add two complex numbers using operator overloading Binary operator overloading

Lesson 69 : **Overloading Equality Operator Comparison Relational C program**

Lesson 70 : **Virtual Functions In C**

Lesson 71 : **Pure Virtual Functions In C Abstract Class In C**

Lesson 72 :
Function Overriding in C Method Overriding in C RUN TIME POLYMORPHISM CPP programming

Lesson 73 :
Function overloading vs overriding Early vs Late Binding Compile vs Run Time Polymorphism

Lesson 74 : **Function Templates In C Programming**

Lesson 75 : **Bubble Sort Using Function Templates In C Programming**

Lesson 76 : **Function Templates with Multiple Parameters In C Programming**

Lesson 77 : **Class Templates In C Programming**

Lesson 78 : **Exception Handling in C try throw and catch keywords in CPP**

Lesson 79 : **Multiple Catch Statements in C Catch All Exceptions in CPP**

Lesson 80 : **Rethrowing an Exception in C**

Lesson 81 : **Specifying Exceptions in C**

Lesson 82 : **Access Specifiers in C Visibility Labels in CPP**

Lesson 83 : **Structures vs Class in C Differences between Structures and Class in CPP versus**

Related courses

React Projects Workshops

Node js back end JavaScript

Python programming language

Social Network Theme UI With Sass

Full Stack React Django

Build a Node js App With Sequelize

for Business Contact
business@mindluster.com