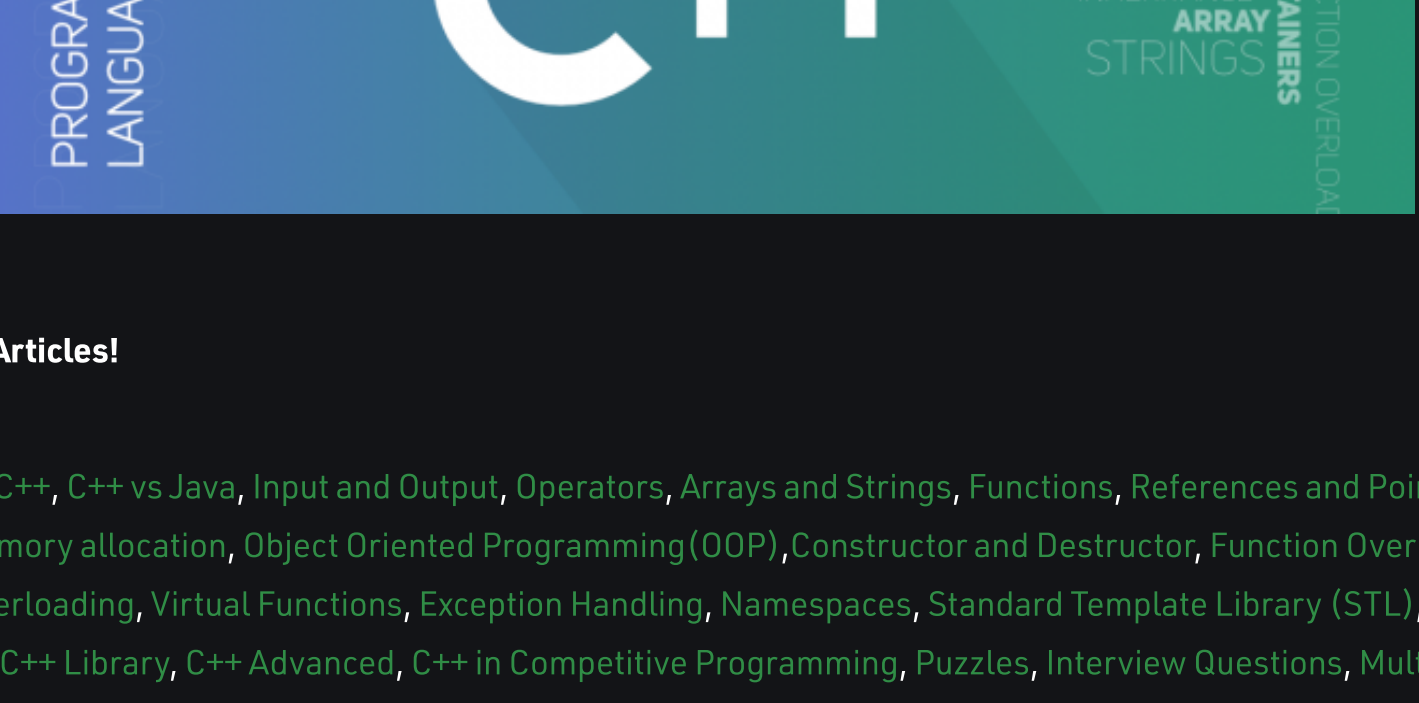


| | |
|---|---|
| Basics | ⤴ |
| C++ Programming Language | |
| Introduction to C++ Programming Language | |
| Setting up C++ Development Environment | |
| C++ Programming Basics | |
| C++ Data Types | |
| Variables in C++ | |
| Loops in C and C++ | |
| Decision Making in C / C++ (if, if-else, Nested if, if-else-if) | |
| Basic Input/Output | ▼ |
| Operators | ▼ |
| Arrays | ▼ |
| TS | ▼ |
| Functions | ▼ |

C++ Programming Language

Last Updated : 17 Jul, 2022

C++ is a general-purpose programming language and is widely used nowadays for competitive programming. It has imperative, object-oriented and generic programming features. C++ runs on lots of platforms like Windows, Linux, Unix, Mac etc.



C++ Recent Articles!

Basics, C vs C++, C++ vs Java, Input and Output, Operators, Arrays and Strings, Functions, References and Pointers, Dynamic memory allocation, Object Oriented Programming (OOP), Constructor and Destructor, Function Overloading, Operator Overloading, Virtual Functions, Exception Handling, Namespaces, Standard Template Library (STL), Inheritance, C++ Library, C++ Advanced, C++ in Competitive Programming, Puzzles, Interview Questions, Multiple Choice Questions

Start Your Coding Journey Now!

Login

Register

- Setting up C++ Development Environment
 - Writing first C++ program (Practice)
 - void main or main()
 - C++ Data Types (Practice)
 - Basic Input/Output
 - Response on exceeding valid range of data types
 - C++ Preprocessors
 - Operators in C++ (Practice)
 - Loops (Practice)
 - Decision Making in C++ (Practice)
 - Execute both if and else simultaneously
 - How to compile 32-bit program on 64-bit gcc in C and C++
 - Switch statement in C++ (Practice)
 - Functions in C++ (Practice)
 - Arrays in C/C++ (Practice)
 - Strings in C++ (Practice)
 - Pointers in C++ (Practice)
 - References in C++
 - Introduction to OOP in C++
- C vs C++**
- C program that won't compile in C++
 - Undefined Behaviour in C and C++
 - Name Mangling and extern "C" in C++
 - void * in C vs C++
 - Program that produces different results in C and C++
 - Type difference of character literals in C vs C++
 - Difference between Structures in C and C++

C++ vs Java

- Inheritance in C++ vs Java
 - static keyword in C++ vs Java
 - default virtual behavior in C++ vs Java
 - Exception Handling in C++ vs Java
 - ForEach in C++ vs Java
 - Templates in C++ vs Generics in Java
 - Floating Point Operations & Associativity in C, C++ and Java
 - Similarities between Java and C++
- Input and output**
- STD Redirection in C++
 - Clearing The Input Buffer
 - Basic Input/Output (Practice)
 - cout << endl vs cout << "\n" in C++
 - Problem with scanf() when there is fgets()/gets()/scanf() after it
 - How to use getline() in C++ when there are blank lines in input?
 - scanf() and fscanf() in C – Simple Yet Powerful
 - Using return value of cin to take unknown number of inputs in C++
 - How to change the output of printf() in main() ?
 - Implementation of a Falling Matrix
 - What does buffer flush means in C++ ?
 - kbhit in C language
 - Code to generate the map of India

Operators

- Operators in C++
- Unary operators in C/C++
- Conditionally assign a value without using conditional and arithmetic operators
- Execution of printf with ++ operators
- Set a variable without using Arithmetic, Relational or Conditional Operator
- Scope Resolution Operator vs this pointer
- Pre-increment (or pre-decrement)
- new and delete operator in C++
- CHAR_BIT in C
- Casting operators | Set 1 (const_cast)

Arrays and Strings

- Arrays in C/C++
- Array of Strings
- Multidimensional arrays in C/C++
- Row string literal
- Counts of distinct consecutive sub-string of length two
- Converting string to number and vice-versa
- Find size of array in C/C++ without using sizeof
- How to quickly reverse a string in C++?
- Tokenizing a string in C++
- getline() function and character array
- Convert string to char array in C++
- C++ string class and its applications, Set 2
- How to create a dynamic 2D array inside a class in C++ ?
- Lexicographically next permutation
- Print size of array parameter
- Split a string in C/C++, Python and Java
- Stringstream in C++ and its applications
- Strchr() function in C/C++
- isspace() in C/C++ and its application to count whitespace characters
- Char* vs std::string vs char[] in C++
- Std::lexicographical_compare() in C++ STL
- Std::string::at in C++
- Std::substr() in C/C++
- Std::stoi() and std::stol() functions in C++
- Extract all integers from string in C++
- Strchr() function in C++ and its applications
- Strncat() vs strncat() in C++
- Strncat() function in C/C++
- Strrchr() in C
- strcoll() in C/C++
- Why strcpy and strncpy are not safe to use?

Functions

- Functions in C++
- Default Arguments
- C function argument and return values
- Inline Functions
- Return from void functions
- Returning multiple values from a function using Tuple and Pair
- Function Call Puzzle
- Factnors
- Ciel and floor functions in C++
- Const member functions
- Std::atoi(), atoi() and atol() functions in C/C++
- swap() in C++
- std::move() function in C++
- std::wscat() function in C++
- std::wscmp() function in C++ with Examples
- std::wncpy() function in C++ with Examples
- std::wcslen() function in C++ with Examples
- std::diff() function in C++
- std::asctime() function in C++
- std::localtime() function in C++
- std::scalbn() function in C++
- std::isunordered() function in C++
- std::isnormal() in C++
- std::isint() function in C++
- std::quick_exit() function in C++ with Examples
- std::clock() function in C/C++
- std::nearbyint() function in C++
- std::quick_exit() function in C++ with Examples
- std::wscmp() function in C++ with Examples
- std::wncpy() function in C++ with Examples
- std::wcslen() function in C++ with Examples

Pointers and References

- Pointers in C and C++
- What is Array Decay in C++? How can it be prevented?
- Opaque Pointer
- References
- Can references refer to invalid location?
- Pass arguments by reference or pointer
- Smart Pointers
- this pointer
- Type of this pointer
- delete this
- std::auto_ptr, unique_ptr, shared_ptr and weak_ptr
- Dangling, Void, Null and Wild Pointers
- Passing by pointer Vs Passing by Reference
- NaN in C++ – What is it and how to check for it?
- nullptr
- Pointers vs References in C++

Dynamic memory allocation

- new and delete operator in C++
- malloc() vs new
- delete() and free()
- Std::get_temporary_buffer in C++

Object Oriented Programming(OOP)

- Object oriented design
- Introduction to OOP in C++
- Classes and Objects
- Access Modifiers
- Inheritance
- Polymorphism
- Encapsulation
- Data Abstraction
- Structure vs class
- Can a C++ class have an object of self type?
- Why is the size of an empty class not zero?
- Static data members in C++
- Some interesting facts about static member functions
- Friend class and function
- Local Classes
- Nested Classes
- Simulating final class

Constructor and Destructor

- Constructors
- Copy Constructor
- Destructors
- Does compiler create default constructor when we write our own?
- When should we write our own copy constructor?
- When is copy constructor called?
- Initialization of data members
- Use of explicit keyword
- When do we use Initialize List in?
- Default Constructors
- Private Destructor
- Playing with Destructors
- Copy elision
- C++ default constructor | Built-in types
- When does compiler create a default constructor and copy constructor?
- Why copy constructor argument should be const in C++?
- Advanced C++ | Virtual Constructor
- Advanced C++ | Virtual Copy Constructor
- When are static objects destroyed?
- Is it possible to call constructor and destructor explicitly?

Function Overloading

- Function Overloading
- Functions that can't be overloaded
- Function overloading and const keyword
- Function overloading and return type
- Does overloading work with Inheritance?
- Can main() be overloaded
- Function Overloading and float

Operator Overloading

- Operator Overloading
- Copy constructor vs assignment operator
- When should we write our own assignment operator?
- Operators that cannot be overloaded
- Conversion Operators
- Is assignment operator inherited?
- Default Assignment Operator and References
- Overloading stream insertion (<<) and extraction (>>) operators
- Overloading array index operator []

Virtual Functions

- Virtual Functions and Runtime Polymorphism
- Default arguments and virtual function
- Virtual functions in derived classes
- Can static functions be virtual?
- Virtual Destructor
- Virtual Constructor
- Virtual Copy Constructor
- RTTI (Run-time type information)
- Can virtual functions be private?
- Inline virtual function
- Pure Virtual Functions and Abstract Classes
- Pure virtual destructor

Exception Handling

- Exception Handling Basics
- Stack Unwinding
- Catching base and derived classes as exceptions
- Catch block and type conversion
- Exception handling and object destruction

Namespaces

- Namespace in C++ | Set 1 (Introduction)
- Set 2 (Declaring namespace and unnamed namespace)
- Namespace in C++ | Set 3 (Accessing, creating header, nesting and aliasing)
- Inline namespaces and usage of the "using" directive inside namespaces
- Can namespaces be nested?

Algorithms

- Introduction to STL
- Sorting
- Searching

Containers:

- Pair (Practice)
- Vector (Practice)
 - Ways to copy a vector in C++
 - Sorting 2D Vector in C++ | Set 3 (By number of columns), (Sort in descending order by first and second)
 - Sorting 2D Vector in C++ | Set 2 (In descending order by row and column)
 - 2D vector in C++ with user defined size
 - Vector::clear() and vector::erase() in C++ STL
 - Passing vector to a function in C++
 - Vector::push_back() and vector::pop_back() in C++ STL
 - Vector::empty() and vector::size() in C++ STL
 - vector::front() and vector::back() in C++ STL
 - Initialize a vector; Different ways
 - Sorting 2D Vector in C++ | Set 1 (By row and column), (Sort by first and second)
 - Computing index using pointers returned by STL functions in C++
- List
 - List in C++ | Set 2 (Some Useful Functions)
 - Forward List in C++ | Set 1 (Introduction and Important Functions)
 - Forward List in C++ | Set 2 (Manipulating Functions)
 - list::remove() and list::remove_if() in C++ STL
 - Forward_list::front() and forward_list::empty() in C++ STL
 - Forward_list::remove() and forward_list::remove_if() in C++ STL
 - forward_list::unique() in C++ STL
 - forward_list::reverse() in C++ STL
 - forward_list::max_size() in C++ STL
 - forward_list::splice_after() in C++ STL
 - list::empty() and list::size() in C++ STL
 - list::front() and list::back() in C++ STL
 - list::pop_front() and list::pop_back() in C++ STL
 - list::push_front() and list::push_back() in C++ STL
 - list::push_front() function in C++ STL
 - list::pop_back() function in C++ STL
 - list::pop_front() function in C++ STL
 - list::reverse function in C++ STL
 - list::resize() function in C++ STL
 - list::size() function in C++ STL
 - list::max_size() function in C++ STL
- Dequeue
- Deque::empty() and deque::size() in C++ STL
- Deque::pop_front() and deque::pop_back() in C++ STL
- Deque::clear() and deque::erase() in C++ STL
- Queue (Practice)
- Queue::front() and queue::back() in C++ STL
-
- Queue::push() and queue::pop() in C++ STL
- queue::empty() and queue::size() in C++ STL
- Priority Queue
- Stack (Practice)
- stack::push() and stack::pop() in C++ STL
- Forward_list::push_front() and forward_list::pop_front() in C++ STL
- stack::top() in C++ STL
- stack::empty() and stack::size() in C++ STL
- Set (Practice)
 - Count number of unique Triangles using STL | Set 1 (Using set)
 - std::istream_iterator and std::ostream_iterator in C++ STL
- Std::next_permutation and prev_permutation in C++
- Std::atol and std::atoll in C++
- Shuffle vs random_shuffle in C++
- Difference between set, multiset, unordered_set, unordered_multiset
- Check if a key is present in a C++ map or unordered_map
- Std::stable_partition in C++
- Valarray slice selector
- Std::memchr in C++
- Std::strncmp() in C++
- stable_sort() in C++ STL
- Std::memcmp() in C++
- Std::memset in C++
- Std::bucket_count and std::bucket_size in unordered_map in C++
- Map of pairs in STL
- Range-based for loop in C++
- Std::includes() in C++ STL
- Std::set_symmetric_difference in C++
- Std::sort_heap in C++
- Map vs unordered_map in C++
- Round() in C++
- Modulus of two float or double numbers
- Multiset
- Map (Practice)
- Heap using STL C++

Multimap

- Multimap in C++ Standard Template Library (STL)
- multimap::find() in C++ STL
- multimap::erase() in C++ STL
- map::emplace() in C++ STL
- multimap::emplace_hint() in C++ STL
- multimap::emplace() in C++ STL
- multimap::count() in C++ STL
- multimap::find() in C++ STL
- multimap::erase() in C++ STL
- multimap::begin() and multimap::end() in C++ STL
- multimap::cbegin() and multimap::cend() in C++ STL
- map::cbegin() and cend() function in C++ STL
- multimap::cbegin() and multimap::cend() in C++ STL
- multimap::size() function in C++ STL
- multimap::lower_bound() function in C++ STL
- multimap::swap() function in C++ STL
- multimap::upper_bound() function in C++ STL
- multimap::max_size() in C++ STL
- multimap::insert() in C++ STL
- multimap::equal_range() in C++ STL

CPP-Math

- sinh() function in C++ STL
- cosh() function in C++ STL
- tanh() function in C++ STL
- acos() function in C++ STL
- asinh() function in C++ STL
- acosh() function in C++ STL
- atanh() function in C++ STL

More:

- sort() in C++ STL
 - Strand sort
 - Type Inference in C++ (auto and decltype)
 - transform() in C++ STL
 - Variaadic function templates in C++
 - Template Specialization
 - Implementing Iterator pattern of a singly linked list
 - Binary Search functions in C++ STL
 - Descending order in Map and Multimap of C++ STL
 - Insertion and Deletion in STL Set C++
 - set::key_comp() in C++ STL
 - set::value_comp() function in C++ STL
 - unordered_set::get_allocator() in C++ STL with Examples
- What all is inherited from parent class in C++?
 - Virtual Functions and Runtime Polymorphism in C++
 - Multiple Inheritance in C++
 - What happens when more restrictive access is given to a derived class method in C++?
 - Object Slicing in C++
 - Hiding of all overloaded methods in base class
 - Inheritance and friendship
 - Simulating final class
- C++ Library**
- <random> file – generator and distributions
 - <array> type manipulation
 - C++ programming and STL facts
 - std::split and std::split in C++
 - std::stoi, std::stoul, std::stold in C++
 - C program to demonstrate fork() and pipe()
 - Complex numbers in C++ | Set 1 Set 2
 - libm() library functions for user input
 - Rename function in C/C++
 - Chrono
 - valarray class
 - Floating Point Manipulation (fmod(), remainder(), remquo(), ... in cmath) (Practice)
 - Character Classification: ctype
 - std::printf() in C library
 - Boost::split in C++ library
 - Modulus of two float or double numbers
 - std::is_trivial function in C++
 - Array sum in C++ STL
 - Div() function in C++
 - Exit() vs _Exit() in C and C++
 - Std::none of in C++
 - std::sprintf() in C++
 - std::isctrl() in C++ and its application to find control characters
 - Std::partition_point in C++
 - Iterator Invalidation in C++
 - Fsetround() and fegetround() in C++ and their application
 - rint(), rintf(), rintl() in C++
 - hypot(), hypotf(), hypotl() in C++
 - std::gslice | Valarray generalized slice selector
 - std::setbuf, std::setw, std::setfill in C++
 - std::strxfrm() in C/C++
 - Set position with seekg() in C++ language file handling
 - std::strr() in C/C++
 - DiffTime() C library function
 - Socket Programming
 - Precision of floating point numbers in C++ (floor(), ceil(), trunc(), round() and setprecision())
 - bibtex++> header file
 - std::string class in C++
 - Merge operations using STL in C++ (merge, includes, set_union, set_intersection, set_difference, ...)
 - std::partition in C++ STL
 - Ratio Manipulations in C++ | Set 1 (Arithmetic), Set 2 (Comparison)
 - numeric header in C++ STL | Set 1 (accumulate() and partial_sum()), Set 2 (adjacent_difference(), inner_product() and iota())
 - Bind function and placeholders
 - Array class
 - Tuples
 - Regex (Regular Expression)
 - Common Subtleties in Vector STLs
 - Understanding constexpr specifier
 - unordered_multiset and its uses
 - unordered_multimap and its application
 - Populating a vector in C++ using fill() and fill_n()
 - Writing OS Independent Code in C/C++
 - C Program to display hostname and IP address
 - Database Connectivity using C/C++
 - C++ bitset and its application
 - unordered_map in STL and its applications
 - unordered_set in STL and its applications
 - nextafter() and nexttoward()

C++ Advanced

- User Defined Literal
 - Placement new operator
 - Advanced C++ with boost library
 - Copy-and-Swap Idiom
 - Zombie and Orphan Processes
 - Lambda expression
 - C++ | Signal Handling
 - Preventing Object Copy in C++
 - Command line arguments in C++
- C++ in Competitive Programming**
- Writing C/C++ code efficiently in Competitive programming
 - Useful Array algorithms in C++ STL
 - Searching in fork()
 - Data Type Ranges and their macros
 - Cin-Cout vs Scanf-Printf
 - getchar_unlocked() – faster input in C/C++ for Competitive Programming
 - std::sort() vs C++ sort()
 - Middle of three using minimum comparisons
 - Check for integer overflow on multiplication
 - Generating Test Cases (generate()) and generate_n()

Interview Questions

- Commonly Asked C++ Interview Questions | Set 1
- Commonly Asked OOP Interview Questions | Set 1
- C/C++ Programs

Quick Links:

- Recent Articles on C++
- Practice Track on C++
- C++ Output & Multiple Choice Questions

GeeksforGeeks Courses

DSA Self Paced

Learn Data Structures and Algorithms with DSA Self-Paced Course trusted by over 75,000 students! Prepare for SDE interviews of top companies such as Microsoft, Amazon, and Adobe, as well as other top product-based companies. Built with years of experience by industry experts and gives you a complete package of video lectures, practice problems, quizzes, discussion forums and contests, learn and master DSA at the best price possible with GeeksforGeeks. [Start Today!](#)

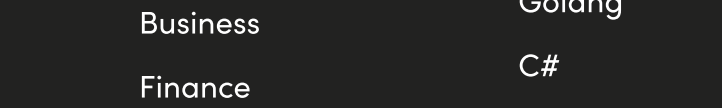
Live Courses

Get best-in-industry Interactive LIVE & Self-Paced Courses with Individual Attention by Leading Experts to upskill yourself and get into your dream company. Clear your doubts to the instructor just like an offline classroom yourself. Check out Valuable Live Courses by GeeksforGeeks to Encourage Out-of-the-box thinking, leading to Clarity in Concepts, Creativity and Innovative Ideas. – [DSA Live for Working Professionals](#), [System Design Live](#), [Competitive Programming Live](#), and more!

Language Foundation Courses[C Programming / C++ / JAVA / Python]

GeeksforGeeks Language Foundation Courses – [C Programming](#) | [Java Foundation](#) | [Python Foundation](#) | [C++ Foundation](#) – Learn any programming language from scratch and master all of its core principles for a strong programming foundation at the lowest price possible. These courses are for complete beginners who want to get started with programming and build their foundations.

Exclusive Hiring Challenge For SDE & SDE 2



Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments