

# Guida alla programmazione in C++

---

Questa guida è per C++ dalla versione C++20 in poi  
utilizzando come IDE Visual Studio 2022

Indice:

## LIVELLO BEGINNER

- I commenti
- Le basi del linguaggio C++
  - Hello World
  - Spiegazione
- L'output (o stampa a video)
- Le sequenze di escape
- La tabella ASCII
- I datatype
  - I datatype primari
  - I **typedef** e gli alias
- Le variabili e le costanti
  - Inizializzazione
  - Cambiare il valore
  - L'ambito delle variabili
- Namespace
- L'input utente
- Gli operatori matematici
  - Gli operatori matematici primari
  - Gli operatori matematici composti
  - Gli operatori unari
- La conversione tra datatype
  - Il type cast
  - Lo static cast
- I concetti di base delle stringhe
  - Output
  - Input
  - Input senza **cin**
- Le istruzioni IF-ELSE
  - Che cos'è
  - Le parentesi graffe
  - Gli operatori
  - Gli IF-ELSE e le eccezioni
  - Gli IF-ELSE e l'assegnazione
- L'operatore ternario
  - Senza assegnazione
  - Con l'assegnazione
- Gli operatori bitwise
- L'istruzione GOTO
- Il ciclo WHILE
- Il ciclo DO-WHILE
- Il ciclo FOR
- Le parole chiave **break** e **continue**

- L'istruzione SWITCH
- Alcune funzioni esterne utili
  - Per gli interi
  - Per i caratteri
- Le costanti matematiche
- Le funzioni matematiche
  - Arrotondamento
  - Valore assoluto
  - Esponenti
  - Logaritmi e trigonometria
- La complessità temporale di un'operazione

## LIVELLO INTERMEDIATE

- Le funzioni
  - Le basi
  - I parametri
  - `return`
  - Passare parametri per riferimento
  - Funzione parametro
  - La ricorsione
  - Dichiarazione e definizione
  - Gli specificatori
  - Gli attributi
- Le direttive
- Le macro
  - Macro come costanti
  - Macro come funzioni
  - Operatori
  - Esempi
- Le enumerazioni
  - `enum`
  - `enum class`
- Le strutture
- Le unioni
- Gli array
  - Inizializzazione di un array
  - Accesso agli elementi
  - Dimensione
  - Passare un array a una funzione
  - Il ciclo FOREACH
  - Array multidimensionali
- Algoritmi degli array
  - Algoritmi di ordinamento
  - Algoritmi di ricerca
- I puntatori
  - Sintassi di base

- Puntatori che puntano a un array \ struttura
- Puntatori nell'heap
- Le eccezioni di `<stdexcept>`
- Puntatori intelligenti
- `Windows.h`
  - Cambiare l'attributo della console
  - Riposizionare il cursore
  - Ottenere i dati della console
  - Nascondere il cursore
  - Pulire un'area dello schermo
  - Inviare un input utente
  - `MessageBox`
- Le stringhe
  - Ottenere la dimensione di una stringa
  - Capire se una stringa è vuota
  - Accesso agli elementi di una stringa
  - Convertire una stringa in un numero
  - Convertire un numero in una stringa
  - Trovare un carattere in una stringa
  - Inserire un carattere in una stringa
  - Tagliare una stringa
  - Aggiungere un carattere a una stringa
  - Concatenare due stringhe
  - Creare una sottostringa
  - Altro
- Gli stringstream
  - Ottenere la stringa di uno stringstream
  - Gli stringstream di input
  - Gli stringstream di output
  - La manipolazione dell'input
  - `std::oct`, `std::dec`, `std::hex`
  - `std::uppercase` e `std::nouppercase`
  - `std::fixed`, `std::scientific` e `std::defaultfloat` con `std::setprecision`
- Le espressioni regolari
  - Le funzioni di ricerca
  - Le regole delle regex
- I vettori
  - Inizializzare un vettore
  - Metodi
- `queue` e `deque`
  - `std::queue`
  - `std::vector` e `std::deque`
- Le mappe
  - Dichiarazione
  - Inserimento
  - Eliminazione

- Capire se la mappa contiene l'elemento
  - Iterare su ogni elemento
  - Differenze
- Misurare il tempo
  - Utilizzo di `std::chrono`
  - Utilizzo delle API di Windows
  - Utilizzo di `GetTickCount`
  - Utilizzo di `clock`
- Generare dei numeri casuali
  - Con la funzione `rand`
  - Con i `random_device`
  - Con il mersenne twister

## LIVELLO ADVANCED

- Le classi
- I costruttori e il distruttore
  - Sintassi dei costruttori
  - Sintassi del distruttore
  - Il costruttore di copia
  - Il costruttore di spostamento
  - Costruttore con `std::initializer_list`
- L'ereditarietà
  - Derivare una classe
  - Ereditare i costruttori
  - Sovrascrivere un metodo
  - Il puntatore `this`
  - Costruttori e distruttori
  - Distruttore virtuale
- Il sovraccarico degli operatori
  - Sovraccarico dell'operatore `=`
  - Sovraccarico degli operatori logici e di confronto
  - Sovraccarico degli operatori aritmetici
  - Sovraccarico degli operatori speciali
- I template
  - Funzioni template
  - Template doppi
  - Classi template
- Tecniche avanzate dei template
  - Template facoltativi e alias
  - Template con valori costanti
  - La specializzazione di un template
  - Template variadici
  - `<type_traits>`
  - Metaprogrammazione
- La parola chiave `friend`
- I thread

- Thread con una funzione
- Thread con una lambda
- Passare parametri a un thread di una lambda
- Thread di una funzione membro
- Passare parametri a un thread di una funzione
- Classi importanti con i thread
  - `std::mutex`
  - `std::condition_variable`
  - `std::future` e `std::promise`
  - `std::atomic`
- La parallelizzazione del codice
  - Con `<execution>`
  - Con `Concurrency`