

LA MODERNA PROGRAMMAZIONE IN C++

Obiettivi

Obiettivo principale del corso è introdurre i concetti fondamentali del C++ moderno, ossia quelli istituiti dallo standard ISO C++11 in avanti.

Questi concetti spaziano dalla semantica dello spostamento degli oggetti alla deduzione automatica dei tipi, passando attraverso l'introduzione delle funzioni lambda e del supporto per il multitasking nelle STL.

Nel corso verrà fatta una panoramica dei concetti fondamentali introdotti con il C++11, C++14 e C++17 e si approfondiranno le conseguenze che hanno avuto sulla semantica della programmazione in C++, incluso l'aggiornamento delle buone prassi del vecchio stile di programmazione in C++98.

Destinatari

Il corso è rivolto al personale che conosce già il linguaggio di programmazione in C++ ed ha un minimo di esperienza nella scrittura di una classe.

Date

Da venerdì 1 a venerdì 8 luglio 2022

Orario

La durata del corso online si prevede di 12 ore, suddivise in 6 lezioni di due ore al giorno.

N. partecipanti

Massimo 20

Responsabile:

Francesco Cafagna (INFN-Bari)

Segreteria organizzativa:

Angelo Ceres - ceres@ba.infn.it

Docenti:

Francesco Cafagna - francesco.cafagna@ba.infn.it

Streaming:

Sì No

Sede:

ONLINE



PROGRAMMA

Introduzione

- L'evoluzione storica del C++: dal C++98 al C++20.
- Per gli impazienti: una rapida carrellata delle novità introdotte dai nuovi standard e l'impatto sul nostro stile di programmazione.

La deduzione automatica dei tipi

- La deduzione automatica dei tipi nei template.
- La deduzione automatica dei tipi nelle definizioni.

La semantica dello spostamento (move) e le rvalue reference

- Copiare e spostare. La semantica dello spostamento e le sue implicazioni.
- rvalue reference e le nuove categorie dei valori.
- Std::move e std::forward. A cosa servono e come si comportano.
- Consigli di uso dello spostamento.

Lambda expression

- Le lambda expression. Introduzione ed uso.
- L'evoluzione delle lambda expression nei vari standard

I template

- L'impatto sui template dei nuovi standard.
- Variadic template e perfect forwarding.
- Cenni sul metaprogramming.

STL

- Una carrellata delle novità introdotte nei nuovi standard.
- Un esempio di impatto del perfect forwarding: la costruzione diretta di oggetti nei contenitori (emplace).
- Concurrency. Cenni ed esempi di uso.
- Filesystem. Cenni ed esempi di uso.
- Chrono. Cenni ed esempi di uso.
- Altre componenti di uso comune: std::variant, std::tuple, etc. etc.

Altre funzionalità del linguaggio base

- Inizializzazione universale.
- Range loop.
- Enum class.
- Structured binding.
- Aggregate extension.
- Una carrellata sulle funzionalità meno utilizzate.

