

Jazyk C++ – Moderní C++, šablony a standardní knihovny

Kód kurzu: CPP_STD

Kurz je navržen pro C++ programátory, kteří se chtějí seznámit s návrhem šablonových funkcí i tříd a s použitím standardních C++ knihoven (STL).

Pobočka	Dnů	Cena kurzu	ITB
Praha	5	25 500 Kč	50
Bratislava	5	1 020 €	50

Uvedené ceny jsou bez DPH.

Termíny kurzu

Datum	Dnů	Cena kurzu	Typ výuky	Jazyk výuky	Lokalita
19.01.2026	5	23 500 Kč	Online	CZ/SK	Online
19.01.2026	5	1 020 €	Online	CZ/SK	Online
16.02.2026	5	25 500 Kč	Prezenční	CZ/SK	GOPAS Praha
16.03.2026	5	1 020 €	Prezenční	CZ/SK	GOPAS Bratislava
13.04.2026	5	25 500 Kč	Prezenční	CZ/SK	GOPAS Brno
25.05.2026	5	23 500 Kč	Online	CZ/SK	Online
25.05.2026	5	1 020 €	Online	CZ/SK	Online
01.06.2026	5	25 500 Kč	Prezenční	CZ/SK	GOPAS Praha

Uvedené ceny jsou bez DPH.

Co Vás naučíme

- Šablony funkcí a princip STL algoritmů
- Použití predikátů a funkčních objektů
- Šablony tříd a princip STL kontejnerů
- Koncept iterátorů
- Vkládací a proudové itrátory
- STL adaptéry
- Automatická správa dynamicky alokované paměti

Požadované vstupní znalosti

- Znalosti v rozsahu kurzu Programovací jazyk C a C++ - objekty v jazyce C++ (CPP_OOP)

Osnova kurzu

Šablony funkcí

- Princip generického programování
- Klíčové slovo templáře
- Rozvinutí šablony, implicitní a explicitní určení parametrů šablony
- Specializace šablony (vyjímky ze šablony)
- Návrh parametrů šablony
- Základní algoritmy copy, accumulate, find, count, min_element, replace,
- reverse

Volitelné entity jazyka C++

- Přetížení operátoru [] - kulaté závorky
- Návrh a použití funkčních objektů a predikátů

GOPAS Praha

Kodaňská 1441/46
101 00 Praha 10
Tel.: +420 234 064 900-3
info@gopas.cz

GOPAS Brno

Nové sady 996/25
602 00 Brno
Tel.: +420 542 422 111
info@gopas.cz

GOPAS Bratislava

Dr. Vladimíra Clementisa 10
Bratislava, 821 02
Tel.: +421 248 282 701-2
info@gopas.sk



Copyright © 2020 GOPAS, a.s.,
All rights reserved

Jazyk C++ – Moderní C++, šablony a standardní knihovny

- Modifikace a přizpůsobování základních algoritmů `for_each`, `transform`,
- `find_if`, `count_if`, `replace_if`, `min_element`, `accumulate`.

Šablony tříd

- Návrh generického kontejneru
- Základní operace s kontejnery
- Koncept iterátoru a použití kontejnerů v algoritmech

Speciální iterátory

- Návrh vkládacího iterátoru
- Použití funkcí `back_inserter`, `front_inserter`, `inserter`
- Proudové iterátory `input_stream_iterator`, `output_stream_iterator`

Základní STL kontejnery

- Kontejner `vector`, jeho struktura a rozhraní
- Alokační strategie vektoru a invalidování iterátorů
- Generování prvků, algoritmus generace
- Zpětné iterátory
- Kontejnery `deque`, `list` a jejich speciální vlastnosti
- Třídění vektoru a seznamu, duplikování STL algoritmu členskou metodou kontejneru
- Odstranění prvků z pole, vektoru či seznamu pomocí `remove`

STL řetězce

- Třída `basic_string`, `string` a `wstring`
- Elementární řetězcové operace
- Neformátované čtení řetězců z proudu `getline`
- Řetězcové proudy v hlavičce

Asociativní kontejnery

- Kontejnery `set`, `multiset`
- Určování třídícího kriteria
- Algoritmy `find`, `count`, `lower_bound`, `upper_bound`
- Množinové operace `set_union`, `set_intersection`, `set_difference`, `set_symmetric_difference`
- Kontejnery `map`, `multimap` a jejich použití
- Operátor `[]` u kontejneru `map`
- Pomocná třída `pair`

Tajemství hlavičkového souboru

- Předdefinované funkční objekty `less`, `greater`, `equal_to`, `plus`, `minus`, `multiply`...
- Vázání parametrů `bind1st`, `bind2nd`
- Adaptéry členských funkcí `mem_fun`, `mem_fun_ref`
- Adaptér pro normální funkce `ptr_fun`

Automatická správa zdrojů / paměti

- Návrhový vzor `smart pointer`
- Návrh, správné a nesprávné použití třídy `auto_ptr`
- Novinky TR1: `shared_ptr` (`bind`, `mem_fn`)

GOPAS Praha

Kodaňská 1441/46
101 00 Praha 10
Tel.: +420 234 064 900-3
info@gopas.cz

GOPAS Brno

Nové sady 996/25
602 00 Brno
Tel.: +420 542 422 111
info@gopas.cz

GOPAS Bratislava

Dr. Vladimíra Clementisa 10
Bratislava, 821 02
Tel.: +421 248 282 701-2
info@gopas.sk



Copyright © 2020 GOPAS, a.s.,
All rights reserved