

C++-expert, taalontwikkelingen (van C++ 11 naar 20)

Praktijkcursus van 3 dagen - 21u

Ref : VEC - Prijs 2024 : € 1 870 excl. BTW

Deze cursus stelt u in staat om de nieuwe functies te assimileren die zijn geïntroduceerd door de nieuwe C++ standaarden. U behandelt de mogelijkheden van functioneel programmeren, beheerst geheugenbeheer met slimme pointers en maakt gebruik van de andere nieuwe functies van de standaard C++ bibliotheek.

PEDAGOGISCHE DOELSTELLINGEN

Na afloop van de opleiding kan de cursist:

Begrijp de nieuwe functies en verbeteringen van de C++ taal

Lambda-uitdrukkingen gebruiken

Prestaties verbeteren met verplaatsingssemantiek

Objecttoewijzing en -vernietiging beheren met slimme aanwijzers

Een toepassing met meerdere threads ontwikkelen

HANDS-ON WORK

Praktische programmeeroefeningen helpen u de verschillende behandelde concepten te begrijpen.

HET PROGRAMMA

laatste update: 09/2024

1) De komst van C++11

- De verschillende C++98, C++03, C++11, C++14, C++17, C++20 standaarden.
 - Wat is er nieuw in C++11 en de doelstellingen van deze standaard. De toekomst van Boost en STL.
 - De kwestie van de compatibiliteit van oude codes.
 - De beschikbaarheid van ontwikkeltools (compilers, debuggers, IDE, enz.).
- Verificatie van de tooling met behulp van de meegeleverde C++11-code.*

2) Taalverbeteringen

- Sterk getypeerde enums.
 - Het sleutelwoord auto en decltype om typen te vereenvoudigen.
 - De interval-gebaseerde lus.
 - C++20 coroutines.
 - Constexpr voor compileertijd evaluatie.
 - C++20 modules.
 - Variadische sjablonen, C++20 schotelconcepten...
- Implementatie van verbeteringen.*

3) Wijzigingen in klassen

- Constructor delegatie en overervingsbeperkingen.
 - De nieuwe semantiek van verplaatsing en de verplaatsingsconstructor.
 - De override, definitieve =delete, =default richtlijnen.
 - Concept van C++20-module
- Klassen maken.*

DEELNEMERS

C++ ontwikkelaars die meer willen weten over de nieuwe functies die gedefinieerd zijn door de C++11 standaard.

VOORAFGAANDE VEREISTEN

Goede kennis van de taal C++. Praktische ervaring met programmeren in deze taal wordt aanbevolen.

VAARDIGHEDEN VAN DE CURSUSLEIDER

De deskundigen die de cursus leiden zijn specialisten op het betreffende vakgebied. Zij werden geselecteerd door onze pedagogische teams zowel om hun vakkennis als hun pedagogische vaardigheden voor elke cursus die zij geven. Zij hebben minstens vijf tot tien jaar ervaring in hun vakgebied en oefenen of oefenden verantwoordelijke bedrijfsfuncties uit.

BEOORDELINGSMODALITEITEN

De cursusleider beoordeelt de pedagogische vooruitgang van de deelnemer gedurende de gehele cursus aan de hand van meerkeuzevragen, praktijksituaties, praktische opdrachten, ... De deelnemer legt ook van tevoren en naderhand een test af ter bevestiging van de verworven kennis.

PEDAGOGISCHE EN TECHNISCHE MIDDELEN

- De gebruikte pedagogische middelen en cursusmethoden zijn voornamelijk: audiovisuele hulpmiddelen, documentatie en cursusmateriaal, praktische oefeningen en correcties van de oefeningen voor praktijkstages, casestudies of reële voorbeelden voor de seminars.
- Na afloop van de stages of seminars verstrekt ORSYS de deelnemers een evaluatievragenlijst over de cursus die vervolgens door onze pedagogische teams wordt geanalyseerd.
- Na afloop van de cursus wordt een presentielijst per halve dag verstrekt, evenals een verklaring van de afronding van de cursus indien de stagiair alle sessies heeft bijgewoond.

TOEGANGSMODALITEITEN EN -TERMIJNEN

De inschrijving dient 24 uur voor aanvang van de cursus plaatsgevonden te hebben.

TOEGANKELIJKHEID VOOR MINDERVALIDEN

Is voor u speciale toegankelijkheid vereist? Neem contact op met mevr. FOSSE, contactpersoon voor mindervaliden, via het adres psh-accueil@ORSYS.fr om uw verzoek en de haalbaarheid daarvan zo goed mogelijk te bestuderen.

4) Draad gebruiken

- Een thread declareren en uitvoeren. Wachten op het einde van de uitvoering met join().
- Gegevens lokaal voor een thread beheren.
- Een resultaat herstellen met future<> en async().

Praktisch werk

5) Hash-gebaseerde unordered_set en unordered_map containers.

- Bereiken met std::view voor on-the-fly evaluatie van C++20.
- Tijdbeheer met de chrono namespace.
- String-opmaak met C++20.

Nieuwe functies implementeren.

6) Functioneel programmeren met lambda-uitdrukkingen

- Declaratie, typering, implementatie en gebruik.

Afsluitingsbeheer, met vastlegging door waarde of door verwijzing van variabelen gekoppeld aan de context.

DATA

Neem contact met ons op