Opleiding : C++, Object programmeren

Praktijkcursus - 5d - 35u00 - Ref. CGE
Prijs : 2610 € V.B.

NEW

Na afloop van de cursus zijn de deelnemers in staat de fundamentele principes van objectgeoriënteerd ontwerpen toe te passen en toepassingen in C++ te ontwerpen. Dit trainingsprogramma is gericht op medewerkers in professionele branches die onder de OPCO Atlas vallen.

Pedagogische doelstellingen

Aan het einde van de training is de deelnemer in staat om:

- De syntaxis en fundamentele concepten van C++ begrijpen
- De belangrijkste aanvullingen op de C++-standaarden onder de knie krijgen
- De principes van objectgeoriënteerd ontwerp toepassen
- Eenvoudige programma's schrijven met behulp van goede ontwikkelpraktijken
- Besturingsstructuren en gegevenstypen gebruiken in C++
- Basisverwerking van bestanden en geheugen

Doelgroep

Voor OPCO Atlas leden: ontwikkelaars, ingenieurs, projectmanagers met nauwe banden met ontwikkeling.

Voorafgaande vereisten

Bekendheid met de principes van objectgeoriënteerd programmeren (OOP) en ervaring met een programmeertaal...

Praktische modaliteiten

Praktisch werk

Casestudies en praktische oefeningen.

Leer methodes

70% pratique – 30% théorie. Pour optimiser le parcours d'apprentissage, des modules e-learning peuvent être fournis avant et après la session présentielle ou la classe virtuelle, sur simple demande du participant.

DEELNEMERS

Voor OPCO Atlas leden: ontwikkelaars, ingenieurs, projectmanagers met nauwe banden met ontwikkeling.

VOORAFGAANDE VEREISTEN

Bekendheid met de principes van objectgeoriënteerd programmeren (OOP) en ervaring met een programmeertaal...

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.

Opleidingsprogramma

1 Java, objectgeoriënteerd leren programmeren - Vooropleiding digitale leerinhoud

- Inleiding.
- Klassen.
- De erfenis.

Digitale activiteiten
Object-georiënteerd programmeren (OOP) is een paradigma dat tegenwoordig in alle moderne programmeertalen te vinden is. Deze concepten maken het mogelijk om code te produceren die efficiënt, krachtig en gemakkelijk te onderhouden is. Deze online cursus introduceert de belangrijkste concepten van objectgeoriënteerd programmeren, zoals het begrip klasse en overerving.

2 Een herinnering aan hoe C++ werkt

- Presentatie van de taal C++ en de ontwikkelingen.
- Hulpmiddelen installeren (compiler, IDE, projectmanager).
- Structuur van een C++-programma.
- Compilatie, uitvoering, beheer van bronbestanden.

Praktisch werk
Installation et configuration de l'environnement. Structure et syntaxe de base. Exercices de contrôle de flux.

3 Tabellen, strings en gegevensbeheer

- Statische en dynamische tabellen.
- Tekenreeksen (C en C++).
- Standaard I/O en bestanden.

Praktisch werk
Manipulation de données. Exercices sur les structures de données. Optimisation et bonnes pratiques.

4 Objectgeoriënteerd programmeren

- Klassen en objecten.
- Inkapseling, abstractie.
- Bouwers, vernietigers.
- Statische en instantie-leden.

Praktisch werk
Création de classes simples. Héritage et polymorphisme. Cas d'usage avancés.

5 Testen en optimaliseren

- Unit testen met C++ frameworks (Catch2, GoogleTest).
- Technieken voor prestatieoptimalisatie.
- Geheugenbeheerstrategieën.

Praktisch werk
Tests unitaires avancés. Optimisation des performances. Déploiement optimisé.

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.

6 Geheugenbeheer in C++

- Dynamische toewijzing (nieuw, verwijderen).
- Pointers, referenties, slimme pointers.
- Geheugenlekken, bronbeheer (RAII).
- Goede geheugenbeheerpraktijken.

Praktisch werk
Manipulation de la mémoire. Atelier RAII et gestion des ressources. Optimisation mémoire.

7 Inleiding tot de STL (Standard Template Library)

- Presentatie van STL en de voordelen ervan.
- Belangrijkste containers: vector, lijst, kaart, set.
- Iteratoren en verzamelpaden.
- Standaard algoritmen (sorteren, vinden, enz.).

Praktisch werk
Manipulation des conteneurs STL. Exercices sur les algorithmes STL. Optimisation et bonnes pratiques STL.

8 Geavanceerde patronen en ontwerp

- Klassieke ontwerppatronen (Singleton, Factory, Observer, enz.).
- Sjablonen gebruiken voor generieke patronen.
- Best practice in objectgeoriënteerd ontwerp.

Praktisch werk
Implémentation de patterns. Patterns avancés et templates. Cas d'usage et revue de code.

9 Geavanceerd testen en optimaliseren

- Geavanceerd unit testen (mocks, geparametriseerde tests).
- Prestatieoptimalisatie (profilering, codeanalyse).
- Refactoringstrategieën.

Praktisch werk
Tests avancés. Optimisation et refactoring. Déploiement et synthèse.

10 Generieke programmering en sjablonen

- Functie- en klassensjablonen.
- Specialisatie en overbelasting door sjablonen.
- Variadische sjablonen en basismetaprogrammering.
- Goede praktijken en te vermijden valkuilen.

Praktisch werk
Création de templates. Métaprogrammation et templates avancés. Optimisation et bonnes pratiques.

11 Uitzonderingsafhandeling en robuustheid

- Uitzonderingsafhandeling (proberen, vangen, gooien).
- Aangepaste uitzonderingen.
- Beste praktijken in foutenbeheer.
- Invloed op prestaties en leesbaarheid.

Praktisch werk
Manipulation des exceptions. Exercices sur la robustesse. Bonnes pratiques et revue de code.

12 Integratie van complexe projecten

- Een project met meerdere bestanden organiseren.
- Gebruik van CMake of andere bouwtools.
- Beheer van afhankelijkheden en modulariteit.
- Documentatie en geautomatiseerde tests.

Praktisch werk
Structuration d'un projet. Intégration et gestion des dépendances. Cas d'usage et revue de projet.

13 Testen, CI/CD en synthese

- Geavanceerde unit- en integratietests.
- Inleiding tot continue integratie (CI/CD).
- Geautomatiseerde bouw- en testtools.
- Samenvatting van de leerstof van de dag.

Praktisch werk
Mise en place de tests automatisés. CI/CD et automatisation. Synthèse et plan d'action.

14 Gevorderd programmeren in C++

- Lambda-uitdrukkingen, auto, nullptr, move-semantiek.
- For-range lussen, uniforme initialisatie.
- Geavanceerde slimme aanwijzers, bronbeheer.
- Anonieme functies en afsluitingen.

Praktisch werk
Exercices sur les nouveautés du langage. Ateliers sur la modernisation du code. Optimisation avancée.

15 Veiligheid en robuustheid in C++

- Geheugenbeveiliging (buffer overflow, use-after-free).
- Goede invoervalidatiepraktijken.
- Gelijktijdig toegangsbeheer (mutex, threads).
- Tools voor beveiligingsanalyse.

Praktisch werk
Analyse de vulnérabilités. Exercices sur la concurrence. Bonnes pratiques et revue de code.

16 Prestaties en multithreading

- Inleiding tot multithreading in C++.
- Gebruik van threads, futures en beloften.
- Synchronisatie en beheer van gedeelde bronnen.
- Hulpmiddelen voor profilering en prestatieanalyse.

Praktisch werk
Mise en œuvre du multithreading. Optimisation de la concurrence. Cas d'usage et revue de code.

17 Testen, bewaking en synthese

- Prestatie- en belastingstesten.
- Monitoringstools (Valgrind, perf, enz.).
- Analyse van logs en detectie van anomalieën.
- Samenvatting van de leerstof van de dag.

Praktisch werk
Tests de performance. Monitoring et analyse. Synthèse et plan d'action.

18 Samenvatting project

- Analyse van specificaties.
- Objectgeoriënteerd en modulair ontwerp.
- Ontwikkeling van een volledige C++-applicatie.
- Integratie van tests, optimalisatie en documentatie.

Praktisch werk
Réalisation du projet. Soutenance et retours.

19 De beste praktijken consolideren

- Goede ontwikkelingspraktijken C++.
- Fout- en uitzonderingsafhandeling.
- Technische documentatie en gebruikersdocumentatie.
- Planning van onderhoud en upgrades.

Praktisch werk
Revue de code croisée. Atelier documentation et maintenance. Synthèse et bonnes pratiques.

20 Persoonlijk actieplan en afsluiting

- Persoonlijke doelen stellen.
- Hulpbronnen en instrumenten voor vooruitgang identificeren.
- Planning voor praktische implementatie.
- Evaluatie en feedback ter plaatse.

Praktisch werk
Élaboration du plan d'action personnel. Évaluation et feedback. Clôture et perspectives.

21 UML, leren modelleren met diagrammen - Post-training digitale

leerinhoud

- Fundamentele concepten.
- Structurele diagrammen.
- Gedragsschema's.

Digitale activiteiten
Deze online training behandelt de grondbeginselen van objectgeoriënteerd ontwerp, de verschillende UML structuur- en gedragsdiagrammen en hun doelstellingen en gebruik. Aan de hand van een ontwerpvoorbeeld brengt de cursus de toepassing van UML in de praktijk om een IT-systeem efficiënt te specificeren, visualiseren en documenteren.

Data en plaats

KLAS OP AFSTAND
2026 : 30 maa., 22 juni, 5 okt., 14 dec.

LILLE
2026 : 30 maa., 22 juni, 5 okt., 14 dec.

PARIS LA DÉFENSE
2026 : 23 maa., 15 juni, 28 sep., 7 dec.