

Nauka programowania w C i C++ (kod: K-CPP)

Opis i cel kursu

Język C++ jest nowoczesnym, wysokopoziomowym językiem programowania, skonstruowanym z myślą o implementowaniu złożonych programów komputerowych. W C++ napisana jest na przykład przeglądarka Mozilla Firefox, baza danych MySQL i środowisko graficzne KDE. Jego inherentną cechą jest wysoka elastyczność, pozwalająca łączyć zaawansowane techniki programistyczne (przeciążanie funkcji, silna obiektowość, szablony i metaprogramowanie, "inteligentne" zarządzanie pamięcią) z niskopoziomowością języka C wtedy, gdy jest ona potrzebna.

Poza praktyczną nauką C++ poruszamy szereg fundamentalnych zagadnień związanych z programowaniem i projektowaniem systemów informatycznych, niezależnych od języka programowania, w tym: obowiązki i usługi systemu operacyjnego, metody ataków oraz obrona przed nimi, refaktoryzacja kodu, podstawy algorytmiki, techniki programowania obiektowego, charakterystyka wygodnych interfejsów użytkownika.

Program

1. Dlaczego C i C++?

- Krótka i niepełna historia języków wysokiego poziomu
- Język kompilowany a interpretowany
- Gdzie, kiedy i dlaczego się programuje w C, ale niekoniecznie w C++
- Czym różni się umiejętność programowania od znajomości języka programowania

2. Fundamentalne pojęcia i działania

- Program a proces
- Interakcja z systemem operacyjnym i środowiskiem wykonania
- Pamięć wirtualna i przestrzeń adresowa
- System plików
- Kodowanie znaków
- Systemy liczbowe

3. Pierwszy program

- Składnia i struktura kodu
- Przestrzeń nazw biblioteki standardowej
- Przebieg wykonania i efekty uboczne
- Etapy przetwarzania kodu źródłowego na plik wykonywalny

4. Zmienne, stałe i ich typy

- Operatory, operacje, działania niezdefiniowane
- Rozmiary typów liczbowych
- Konwersje i rzutowania
- Elementarna interakcja z użytkownikiem
- Problem walidacji danych
- Po co te formalizmy?

5. Przepływ sterowania

- Operacje logiczne, reprezentacja prawdy i fałszu
- Instrukcje warunkowe
- Pętle
- Zmienne lokalne, zakresy widoczności nazw
- Podstawowe algorytmy

6. Model pamięci

Zapytaj o szczegóły

tel. 22 63 64 164
akademia@alx.pl

Najbliższe terminy

2020-06-06 (Zdalnie)
2020-08-22 (Warszawa)
2020-08-22 (Zdalnie)
2020-09-15 (Zdalnie)
2020-09-15 (Warszawa)

- Reprezentacja i położenie zmiennych w pamięci komputera
- Segment kodu, danych, stosu i sterty
- Systemowe mechanizmy ochrony pamięci
- 7. **Tablice, wskaźniki, napisy**
 - Tablice jedno- i wielowymiarowe, organizacja w pamięci
 - Dualizm tablica-wskaźnik
 - Napisy w C a napisy w C++
 - Typowe błędy, ataki rodzaju buffer overflow
 - Arytmetyka wskaźników
- 8. **Podstawowe techniki programistyczne**
 - Uogólnianie programu - podprogramy i funkcje
 - Dynamiczny przydział pamięci
 - Sposoby przekazywania parametrów do/wyników z funkcji
 - Operacje plikowe
- 9. **Koncepcja programowania obiektowego**
 - Klasa, obiekt, metoda, atrybut, stan
 - Interfejs a implementacja
 - Ochrona stanu obiektu
 - Const-correctness
 - Metody i atrybuty klasowe (statyczne)
- 10. **Zarządzanie zasobami w modelu obiektowym**
 - Konstruktory i destruktory
 - Wartość obiektu a jego tożsamość
 - Kopiowanie obiektów - kopie płytkie i głębokie
- 11. **Mechanizm dziedziczenia**
 - Motywacja praktyczna
 - Problem koła i elipsy
 - Konstrukcja i destrukcja obiektów klas pochodnych
 - Polimorfizm i metody wirtualne
 - Metody i klasy abstrakcyjne
 - Wirtualne destruktory
 - Dziedziczenie vs składanie
- 12. **Obsługa wyjątków**
 - Przepływ sterowania programu po wystąpieniu wyjątku
 - Generowanie i przechwytywanie wyjątków
 - Najczęściej spotykane wyjątki z biblioteki standardowej
- 13. **Projekty wieloplikowe**
 - Jednostka kompilacji
 - Deklaracje i definicje, pliki nagłówkowe
 - Reguła jednej definicji
 - Linkowanie, widoczność symboli
 - Biblioteki statyczne i dynamiczne
 - Automatyzacja budowania projektu
- 14. **Szablony**
 - Szablony w funkcjach
 - Szablony w klasach
 - Specjalizacje szablonów
 - Problemy z linkowaniem
 - Wstęp do metaprogramowania
- 15. **Standardowa biblioteka szablonów (STL)**
 - Kolekcje, kolejki, algorytmy, funkcje wyższego rzędu
 - Dobór właściwej struktury danych do problemu i wpływ na wydajność programu
 - Analiza kosztu czasowego i pamięciowego

Zapytaj o szczegóły

tel. 22 63 64 164
akademia@alx.pl

Najbliższe terminy

2020-06-06 (Zdalnie)
2020-08-22 (Warszawa)
2020-08-22 (Zdalnie)
2020-09-15 (Zdalnie)
2020-09-15 (Warszawa)

- Dostosowanie własnych klas do współpracy z generycznymi algorytmami i kolekcjami
16. **Wstęp do programowania w Qt**
- Środowisko pracy z bibliotekami Qt
 - Programowanie oparte o obsługę zdarzeń
 - Generowanie i przekazywanie zdarzeń: sygnały i sloty
 - Projektowanie interfejsu aplikacji graficznej
 - Model obiektów Qt, automatyzacja zarządzania pamięcią

Zapytaj o szczegóły

tel. 22 63 64 164
akademia@alx.pl

Przeznaczenie i wymagania

Od uczestników wymagana jest znajomość ogólnej obsługi komputera (kopiowanie plików, korzystanie z edytora tekstu) i umiejętność sprawnego korzystania z klawiatury.

C/C++ to potężne narzędzia. Kurs prowadzony jest od zera, nie wymaga umiejętności programowania, natomiast jest wskazane, aby słuchacze interesowali się (choćby w przeszłości?) przedmiotami ścisłymi/przyrodniczymi, zagadnieniami popularnonaukowymi, informatyką, być może stawiali pierwsze samodzielne kroki związane z czymkolwiek z obszaru programowania, albo np. serwisowymi.

Programowanie, szczególnie na początku, jest trudne, a jego nauka jest porównywalna z nauką obcego języka lub gry na instrumencie muzycznym: warunkiem skutecznego uczestnictwa w kursie jest zarezerwowanie odpowiedniej ilości czasu w domu na ćwiczenia.

Najbliższe terminy

2020-06-06 (Zdalnie)
2020-08-22 (Warszawa)
2020-08-22 (Zdalnie)
2020-09-15 (Zdalnie)
2020-09-15 (Warszawa)

Certyfikaty

Uczestnicy szkolenia otrzymują imienne certyfikaty sygnowane przez ALX.