



UNIwersYTET
IM. ADAMA MICKIEWICZA
W POZNANIU

Języki programowania Sylabus zajęć

Informacje podstawowe

Kierunek studiów Technologie komputerowe Specjalność - Jednostka organizacyjna Wydział Fizyki i Astronomii Poziom studiów studia inżynierskie pierwszego stopnia Forma studiów studia stacjonarne Profil studiów profil ogólnoakademicki		Cykl dydaktyczny 2024/25 Kod zajęć 04TKOS.32N.04996.24 Języki wykładowe polski Obligatoryjność Obowiązkowy Blok zajęciowy Przedmioty nieprzypisane
Koordynator zajęć	Kamil Szkarłat	
Prowadzący zajęcia	Kamil Szkarłat	
Okres Semestr 2	Forma zajęć / liczba godzin / forma zaliczenia - Wykład: 15, Zaliczenie z oceną - Laboratorium: 45, Zaliczenie z oceną	Liczba punktów ECTS 4

Cele kształcenia dla zajęć

Kod	Cel
C1	Zapoznanie studentów z podstawami programowania z wykorzystaniem języków C i C++.
C2	Opanowanie przez studentów umiejętności pisania kodów źródłowych w językach C (strukturalnym) i C++ (obiektywnym).
C3	Zapoznanie studentów z odpowiednim wykorzystaniem operacji wejścia/wyjścia.
C4	Nauczenie studentów odpowiedniego alokowania pamięci (w tym w sposób dynamiczny).
C5	Nauczenie studentów pracy z tablicami, macierzami oraz wskaźnikami.
C6	Zapoznanie studentów z programowaniem obiektywnym. Przedstawienie klas, ich składowych oraz budowy.
C7	Zapoznanie studentów z dziedziczeniem. Nauczenie umiejętności programowania różnych metod dziedziczenia.

Wymagania wstępne

Wymagana jest wstępna znajomość z kursu Wstęp do programowania, tj. podstawowa wiedza i umiejętności programowania języka proceduralnego w składni języka C z wykorzystaniem tworzenia zmiennych, funkcji, pętli i instrukcji sterujących.

Efekty uczenia się dla zajęć

Kod	Efekty uczenia się dla zajęć w zakresie	Efekty uczenia się dla kierunku	Metody weryfikacji osiągnięcia efektów uczenia się dla zajęć
Wiedzy - Student/ka:			
W1	Student/ka wie jakie są elementarne instrukcje wykonywane przez procesory.	TKO_K3_W01, TKO_K3_W02_inz	Kolokwium pisemne, Kolokwium ustne
W2	Student/ka wie co to są języki wysokiego poziomu proceduralne i obiektywne.	TKO_K3_W02_inz, TKO_K3_W03_inz	Kolokwium pisemne, Kolokwium ustne
W3	Student/ka wie co to są tablice, wskaźniki, łańcuchy znakowe (stringi) oraz programowanie dynamiczne.	TKO_K3_W04_inz, TKO_K3_W05_inz	Kolokwium pisemne, Kolokwium ustne
W4	Student/ka wie co to są klasy, ich budowa oraz warianty w programowaniu obiektywnym.	TKO_K3_W04_inz, TKO_K3_W05_inz	Kolokwium pisemne, Kolokwium ustne
Umiejętności - Student/ka:			
U1	Student/ka potrafi pisać i kompilować proste programy z wykorzystaniem składni języka C oraz C++.	TKO_K3_U03_inz, TKO_K3_U04_inz	Projekt, Zaliczenie praktyczne (analiza wykonawstwa)
U2	Student/ka potrafi pisać programy (w językach C i C++) używając różnych typów danych, tablic, struktur i wskaźników.	TKO_K3_U03_inz, TKO_K3_U04_inz	Projekt, Zaliczenie praktyczne (analiza wykonawstwa)
U3	Student/ka potrafi pisać programy z dynamiczną alokacją pamięci.	TKO_K3_U03_inz, TKO_K3_U04_inz	Projekt, Zaliczenie praktyczne (analiza wykonawstwa)
U4	Student/ka potrafi definiować klasy, odpowiednio używać zdefiniowane przez siebie typy danych, odpowiednio kodować zdefiniowane problemy z wykorzystaniem programowania obiektywnego.	TKO_K3_U03_inz, TKO_K3_U04_inz	Projekt, Zaliczenie praktyczne (analiza wykonawstwa)

Kod	Efekty uczenia się dla zajęć w zakresie	Efekty uczenia się dla kierunku	Metody weryfikacji osiągnięcia efektów uczenia się dla zajęć
U5	Student/ka potrafi używać w programowaniu dziedziczenia wielokrotnego oraz funkcji wirtualnych.	TKO_K3_U03_inz, TKO_K3_U04_inz	Projekt, Zaliczenie praktyczne (analiza wykonawstwa)
Kompetencji społecznych - Student/ka:			
K1	Student/ka rozumie potrzebę uczenia się i doskonalenia swoich umiejętności przez całe życie, jednocześnie potrafi inspirować i organizować proces uczenia się innych osób	TKO_K3_K01, TKO_K3_K02	Kolokwium ustne, Wypowiedź ustna
K2	Student/ka ma wiedzę z zakresu własności intelektualnej (własność kodów źródłowych).	TKO_K3_K03	Kolokwium ustne, Wypowiedź ustna
K3	Student/ka ma umiejętność identyfikacji różnych metod programowania, a jednocześnie posiada świadomość skutków wadliwie działających programów i systemów informatycznych.	TKO_K3_K04	Kolokwium ustne, Wypowiedź ustna

Treści programowe dla zajęć

Lp.	Treści programowe dla zajęć	Efekty uczenia się dla zajęć	Formy zajęć
1.	Informacje ogólne o językach programowania. Kompilacja. Identyfikatory proste (wbudowane), typy danych oraz ich reprezentacja binarna. Zakres widoczności zmiennych. Podstawowe operatory.	W1, U1, K1	Wykład, Laboratorium
2.	Instrukcje sterujące, instrukcje warunkowe oraz pętle. Formatowanie wydruków. Użycie funkcji.	W1, U1, K1	Wykład, Laboratorium
3.	Tablice, macierze, wskaźniki, łańcuchy znakowe (stringi).	W3, U2, K1	Wykład, Laboratorium
4.	Arytmetyka wskaźników i dynamiczna alokacja pamięci.	W3, U2, U3, K1	Wykład, Laboratorium
5.	Struktury i Unie w ujęciu programowania proceduralnego a klasy w ujęciu programowania obiektowego	W2, U4	Wykład, Laboratorium
6.	Wstęp do programowania obiektowego: klasy, składowe klasy, konstruktory i destruktory, funkcje stałe, deklaracje i definicje klasy, zagnieżdżanie klas.	W2, W4, U4, K1	Wykład, Laboratorium
7.	Analiza i programowanie proceduralne oraz zorientowane obiektowo. Różnice w kodach źródłowych. Pojęcie własności intelektualnej w programowaniu (własność kodów źródłowych a plików wykonywalnych).	W2, U1, K2, K3	Wykład, Laboratorium
8.	Referencje, przeciążanie funkcji i konstruktorów. Funkcje zaprzyjaźnione.	W4, U4, K1	Wykład, Laboratorium
9.	Dziedziczenie i dziedziczenie wielokrotne. Dostęp do składowych klas bazowych i potomnych. Wirtualne klasy bazowe.	W4, U5, K1	Wykład, Laboratorium
10.	Operacje I/O w język C++. Manipulatory, Operatory << oraz >>.	W4, U4, K1	Wykład, Laboratorium

Informacje dodatkowe

Forma zajęć	Metody i formy prowadzenia zajęć
Wykład	Wykład z prezentacją multimedialną wybranych zagadnień, Dyskusja, Metoda aktywizująca - "burza mózgów"
Laboratorium	Rozwiązywanie zadań (np.: obliczeniowych, artystycznych, praktycznych), Metoda laboratoryjna, Metoda projektu

Forma zajęć	Warunki zaliczenia zajęć
Wykład	Warunkiem zaliczenia zajęć jest pozytywna ocena z kolokwium. Kryteria oceniania: 51-60% punktów: ocena 3 61-70% punktów: ocena 3,5 71-80% punktów: ocena 4,0 81 - 90% punktów: ocena 4,5 powyżej 91% punktów: ocena 5,0
Laboratorium	Warunkiem zaliczenia zajęć jest pozytywna ocena, na którą składa się: - bieżące przygotowanie do zajęć i aktywność: 40% - realizacja programów na ćwiczeniach i w domu: 30% - prezentacja i zaliczenie projektów programistycznych: 30%

Literatura

Obowiązkowa

1. Kernighan B. W., Ritchie D. M.: "Język ANSI C.", Warszawa, WNT 1988
2. Delannoy C.: "Ćwiczenia z języka C.", Warszawa, WNT 1993
3. Prata S., Język C. Szkoła programowania, Helion 2016
4. Stroustrup B.: "Programowanie: teoria i praktyka z wykorzystaniem C++", Helion 2020
5. Grębosz J.: "Opus Magnum C++11 : programowanie w języku C++", Helion 2020
6. Zieliński J.: "Podstawy programowania w języku C++", Oficyna Wydawnicza "Impuls", 2013

Dodatkowa

1. Grębosz J. "Symfonia C++, tomy 1-3", Oficyna Kallimach 2000
2. Plauger P.J.: "Biblioteka standardowa C++", WNT Warszawa 1997
3. Grębosz J.: "Pasja C++", Oficyna Kallimach 2004

Nakład pracy studenta i punkty ECTS

Rodzaje zajęć studenta	Średnia liczba godzin* przeznaczonych na zrealizowane rodzaje zajęć
Wykład	15
Laboratorium	45
Czytanie wskazanej literatury	20
Przygotowanie projektu	30
Przygotowanie do egzaminu	10

Łączny nakład pracy studenta	Liczba godzin 120
Liczba punktów ECTS	ECTS 4

* godzina (lekcyjna) oznacza 45 minut

Efekty uczenia się dla kierunku

Kod	Treść
TKO_K3_K01	Absolwent/ka jest gotów/gotowa do formułowania pytań służących pogłębieniu własnego zrozumienia określonej tematyki
TKO_K3_K02	Absolwent/ka jest gotów/gotowa do korzystania z opinii ekspertów podczas rozwiązywania problemów teoretycznych oraz praktycznych
TKO_K3_K03	Absolwent/ka jest gotów/gotowa do docenienia istoty uczciwości intelektualnej we własnych i cudzych działaniach, a także do postępowania etycznego
TKO_K3_K04	Absolwent/ka jest gotów/gotowa do pełnienia roli zawodowej informatyka, mając świadomość nieustannych zmian stosowanych paradygmatów i technologii
TKO_K3_U03_inz	Absolwent/ka potrafi zastosować wiedzę matematyczną do formułowania, modelowania, analizy i rozwiązywania elementarnych zadań związanych z fizyką oraz informatyką
TKO_K3_U04_inz	Absolwent/ka potrafi kodować, uruchamiać i testować programy na różnych platformach i w różnych środowiskach programistycznych
TKO_K3_W01	Absolwent/ka zna i rozumie fundamentalne pojęcia i problemy definiujące podstawy dyscyplin fizyka i informatyka
TKO_K3_W02_inz	Absolwent/ka zna i rozumie podstawowe pojęcia i zagadnienia związane z wybranymi głównymi obszarami fizyki i informatyki
TKO_K3_W03_inz	Absolwent/ka zna i rozumie właściwe dla wybranych obszarów zastosowań technologie, narzędzia i urządzenia informatyczne, oraz fizyczne podstawy ich działania
TKO_K3_W04_inz	Absolwent/ka zna i rozumie fundamentalne pojęcia, struktury oraz procesy związane z językami programowania, inżynierią programowania i fizyką komputerową
TKO_K3_W05_inz	Absolwent/ka zna i rozumie fundamentalne pojęcia i zagadnienia związane z projektowaniem i weryfikacją algorytmów oraz struktur danych