



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez
Unię Europejską w ramach
Europejskiego Funduszu
Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



Nazwa przedmiotu		Kod ECTS	
Języki programowania		11.3.1100	
Nazwa jednostki prowadzącej przedmiot			
Instytut Informatyki			
Studia			
wydział	kierunek	poziom	pierwszego stopnia
Wydział Matematyki, Fizyki i Informatyki	Informatyka	forma	stacjonarne
		moduł	wszystkie
		specjalnościowy	wszystkie
		specjalizacja	wszystkie
Nazwisko osoby prowadzącej (osób prowadzących)			
dr inż. Emilia Lubecka; dr hab. Marcin Ciecholewski			
Formy zajęć, sposób ich realizacji i przypisana im liczba godzin		Liczba punktów ECTS	
Formy zajęć		7 Przedmiot w wymiarze 30h wykładu, 30h lab. + praca własna studenta	
Wykład, Ćw. laboratoryjne			
Sposób realizacji zajęć			
zajęcia w sali dydaktycznej			
Liczba godzin			
Ćw. laboratoryjne: 30 godz., Wykład: 30 godz.			
Termin realizacji przedmiotu			
2019/2020 letni			
Status przedmiotu		Język wykładowy	
obowiązkowy		polski	
Metody dydaktyczne		Forma i sposób zaliczenia oraz podstawowe kryteria oceny lub wymagania egzaminacyjne	
<ul style="list-style-type: none"> - Analiza zdarzeń krytycznych (przypadków) - Programowanie na żywo (life coding) - Rozwiązywanie zadań - Wykonywanie doświadczeń - Wykład problemowy 		Sposób zaliczenia	
		<ul style="list-style-type: none"> - Zaliczenie na ocenę - Egzamin 	
		Formy zaliczenia	
		<ul style="list-style-type: none"> - egzamin pisemny testowy - ustalenie oceny zaliczeniowej na podstawie ocen cząstkowych otrzymywanych w trakcie trwania semestru 	
		Podstawowe kryteria oceny	
		<p>Ocenie podlega rozumienie zasad poprawnego programowania oraz stopień znajomości języka C++ oraz Python.</p> <p>Przedmiot kończy się egzaminem pisemnym w formie testu, z którego należy uzyskać min 50%.</p> <p>Aby przystąpić do egzaminu, trzeba zaliczyć ćwiczenia laboratoryjne - również na min 50%</p> <p>Na ćwiczeniach laboratoryjny zdobywa się oceny cząstkowe na podstawie zaimplementowanych programów komputerowych.</p>	
Sposób weryfikacji założonych efektów kształcenia			

zakładany efekt kształcenia	egzamin	kolokwium	projekt	referat	raport	aktywność w dyskusji	obserwacja
K_W0	x						
K_W0		x					
K_U0		x					
K_U0	x	x					

Określenie przedmiotów wprowadzających wraz z wymogami wstępnymi

A. Wymagania formalne

Brak.

B. Wymagania wstępne

Rozumienie podstawowych konstrukcji programistycznych, takich jak instrukcje warunkowe, pętle, funkcje, rekurencja
Znajomość języka C w stopniu średniozaawansowanym (podstawowe typy danych, tablice, struktury, wskaźniki)
Znajomość podstawowych zasad poprawnego programowania strukturalnego i dobrych praktyk programistycznych

Cele kształcenia

Celem przedmiotu jest zapoznanie słuchaczy z zasadami programowania strukturalnego i obiektowego w połączeniu z dobrymi praktykami programistycznymi na bazie wybranych języków programowania: C++ i Python.

Treści programowe

Kurs obejmuje język C++ oraz Python.
Wprowadzenie podstawowych paradygmatów programowania obiektowego.
Podstawowe typy danych, abstrakcja danych - własne typy, hierarchia (dziedziczenie), rzutowanie, polimorfizm.
Typowe składowe programów obiektowych: klasa, obiekt, pole, metoda, konstruktor, itd.
Przeciążanie operatorów, metod i funkcji.
Operacje na strumieniach.
Referencje. Dynamiczne zarządzanie pamięcią
Biblioteki: standardowa, matematyczna, operująca na napisach. Tworzenie własnych modułów
Błędy w programowaniu.
Praca z IDE.

Wykaz literatury

Bruce Eckel, *Thinking in C++*, 2ed (wydanie angielskie lub polskie)
Bjarne Stroustrup, *The C++ Programming Language*, 4ed (wydanie angielskie)
Karol Kuczumski, *Kurs C++ (Od zera do kier koder)*, tutorial (wydanie polskie - na Licencji GNU Wolnej Dokumentacji)
Mark Lutz, *Python. Wprowadzenie* (Helion, wydanie IV, j. polski)
Python programming language, tutorial (wydanie angielskie)
Python 3, tutorialspoint (j. angielski)

Kierunkowe efekty kształcenia

Wiedza

- Znajomość języka C++ i Python w stopniu średniozaawansowanym
- Znajomość podstawowych typów i struktur danych używanych we współczesnych językach programowania
- Znajomość uniwersalnych konstrukcji programistycznych
- Znajomość zasad poprawnego programowania i dobrych praktyk programistycznych

Umiejętności

- Sprawne posługiwanie się językiem C++ i Python
- Umiejętne stosowanie zasad programowania obiektowego
- Dobór odpowiednich dla problemu konstrukcji programistycznych i abstrakcji danych
- Używanie na poziomie podstawowym zintegrowanych środowisk programistycznych (IDE)

Kompetencje społeczne (postawy)

Kontakt

elubecka@inf.ug.edu.pl