



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		11.3.1538	
Nazwa jednostki prowadzącej przedmiot			
Instytut Informatyki			
Studia			
wydział	kierunek	poziom	pierwszego stopnia
Wydział Matematyki, Fizyki i Informatyki	Informatyka	forma	niestacjonarne (zaoczne)
		moduł	wszystkie
		specjalnościowy	wszystkie
		specjalizacja	wszystkie
Nazwisko osoby prowadzącej (osób prowadzących)			
dr Maciej Dziemiańczuk; dr hab. Tomasz Dzido; mgr Mateusz Miotk; mgr Łukasz Mielewczyk; dr inż. Arkadiusz Mirakowski; dr Andrzej Borzyszkowski; dr Adam Kostulak			
Formy zajęć, sposób ich realizacji i przypisana im liczba godzin		Liczba punktów ECTS	
Formy zajęć		12 Przedmiot w wymiarze 40h wykładu i 40h lab. + praca własna studenta	
Wykład, Ćw. laboratoryjne			
Sposób realizacji zajęć			
zajęcia w sali dydaktycznej			
Liczba godzin			
Ćw. laboratoryjne: 40 godz., Wykład: 40 godz.			
Termin realizacji przedmiotu			
2022/2023 zimowy			
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"> - Wykład z prezentacją multimedialną - ćwiczenia na pracowni komputerowej – rozwiązywanie zadań 		Sposób zaliczenia	
		<ul style="list-style-type: none"> - Zaliczenie na ocenę - Egzamin 	
		Formy zaliczenia	
		<ul style="list-style-type: none"> - egzamin pisemny z pytaniami (zadaniami) otwartymi - egzamin pisemny testowy 	
		Podstawowe kryteria oceny	
		Ocena z ćwiczeń na podstawie wyników z prac domowych oraz aktywności na zajęciach. Ocena z wykładu na podstawie wyników testu sprawdzającego rozumienie fragmentów kodu.	
Sposób weryfikacji założonych efektów uczenia się			

zakładany efekt kształcenia	egzamin	projekt	sprawdzian	obserwacja postawy studenta
Wiedza				
K_W05	x		x	
P_W1	x		x	
P_W2	x		x	
Umiejętności				
K_U06		x	x	
P_U09		x	x	
P_U1		x	x	
P_U2		x	x	
P_U3		x	x	
Kompetencje				
K_K01				x
P_K1				x

Określenie przedmiotów wprowadzających wraz z wymogami wstępnymi

A. Wymagania formalne

Brak.

B. Wymagania wstępne

Umiejętność obsługi komputera i instalacji programów.

Cele kształcenia

Celem jest nauczenie studentów programowania w języku C oraz czytania ze zrozumieniem kodu C napisanego przez innych programistów.

Treści programowe

1. *Wprowadzenie do programowania w języku C.* Uruchamianie prostych programów. Składnia i semantyka programu.
2. *Środowisko programistyczne w systemie UNIX.* Kompilacja i konsolidacja programu z użyciem kompilatora gcc. Program make. Debugger gdb.
3. *Typy, operatory i wyrażenia.* Deklaracje. Konwersja typów. Wyrażenia warunkowe. Kolejność obliczeń.
4. *Instrukcje sterujące.* Instrukcje i bloki.
5. *Funkcje i struktura programu.* Zasięg nazw. Pliki nagłówkowe. Preprocesor języka C. Kompilacja warunkowa.
6. *Wskaźniki i tablice.* Argumenty wywołania programu. Wskaźniki do funkcji. Skomplikowane deklaracje.
7. *Struktury.* Deklaracje typedef. Unie.
8. *Wejście i wyjście.* Omówienie standardowych bibliotek: wejścia/wyjścia, funkcji działających na napisach, funkcji matematycznych. Wprowadzenie do tzw. locales. Kodowanie UTF-8.
9. *Abstrakcyjne typy danych i ich implementacja.* Stosy i kolejki. Drzewa poszukiwań binarnych. Tablice dynamiczne.

Wykaz literatury

1. Brian W. Kernighan, Dennis M. Ritchie. Język ANSI C. WNT
 2. Stephen Prata. Język C. Szkoła programowania, Helion
 3. Keith Cooper, Linda Torczon. Engineering a Compiler, MORGAN KAUFMANN
- Podręczniki online:
1. Zed A. Shaw. Learn C The Hard Way
 2. Mike Banahan, Declan Brady, Mark Doran. The C Book
 3. Richard M. Stallman, Roland McGrath, Paul D. Smith. GNU Make Manual
 4. Sandra Loosemore i Richard M. Stallman, Roland McGrath, Andrew Oram, Ulrich Drepper. The GNU C Library

Kierunkowe efekty uczenia się

K_W05: ma ogólną wiedzę na temat różnych paradygmatów programowania i języków programowania
 K_U06: potrafi projektować, tworzyć, uruchamiać i testować programy przy wykorzystaniu dedykowanych narzędzi oraz adekwatnych wzorców
 K_U09: potrafi oceniać przydatność paradygmatów i narzędzi programistycznych do rozwiązywania problemów różnego typu
 K_K01: zna ograniczenia własnej wiedzy i rozumie potrzebę dalszego uczenia się

Wiedza

- Student:
- P_W1: Odróżnia błędy składniowe w kodzie programów od błędów logicznych. Rozumie krótkie fragmenty kodu. Zna podstawowe konstrukcje języka C (K_W05)
 - P_W2: Wie jakie są biblioteki standardowe języka i co zawierają (K_W05)

Umiejętności

- Student:
- P_U1: umie napisać program w C rozwiązujący konkretny problem (K_U06)
 - P_U2: umie zaprogramować w C prosty algorytm (K_U06)
 - P_U3: w swoich programach potrafi skorzystać z gotowych bibliotek i cudzego

	kodu (K_U09)
	Kompetencje społeczne (postawy)
	Student: <ul style="list-style-type: none">• P_K1: rozumie potrzebę nauki innych języków programowania (K_K01)
Kontakt	
maciej.dziemianczuk@inf.ug.edu.pl	