



**KAPITAŁ LUDZKI**  
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez  
Unię Europejską w ramach  
Europejskiego Funduszu  
Społecznego

**UNIA EUROPEJSKA**  
EUROPEJSKI  
FUNDUSZ SPOŁECZNY



<b>Nazwa przedmiotu</b>		<b>Kod ECTS</b>	
Języki programowania		11.3.1538	
<b>Nazwa jednostki prowadzącej przedmiot</b>			
Instytut Informatyki			
<b>Studia</b>			
<b>wydział</b>	<b>kierunek</b>	<b>poziom</b>	<b>pierwszego stopnia</b>
Wydział Matematyki, Fizyki i Informatyki	Informatyka	<b>forma</b>	niestacjonarne (zaoczne)
		<b>moduł</b>	wszystkie
		<b>specjalnościowy</b>	wszystkie
		<b>specjalizacja</b>	wszystkie
<b>Nazwisko osoby prowadzącej (osób prowadzących)</b>			
dr Maciej Dziemiańczuk; mgr Mateusz Miotk; dr Andrzej Borzyszkowski; dr hab. Tomasz Dzido; dr Adam Kostulak; dr inż. Arkadiusz Mirakowski; mgr Łukasz Mielewczyk			
<b>Formy zajęć, sposób ich realizacji i przypisana im liczba godzin</b>		<b>Liczba punktów ECTS</b>	
<b>Formy zajęć</b>		12 Przedmiot w wymiarze 40h wykładu i 40h lab. + praca własna studenta	
Wykład, Ćw. laboratoryjne			
<b>Sposób realizacji zajęć</b>			
zajęcia w sali dydaktycznej			
<b>Liczba godzin</b>			
Wykład: 40 godz., Ćw. laboratoryjne: 40 godz.			
<b>Termin realizacji przedmiotu</b>			
2021/2022 zimowy			
<b>Status przedmiotu</b>		<b>Język wykładowy</b>	
obowiązkowy		polski	
<b>Metody dydaktyczne</b>		<b>Forma i sposób zaliczenia oraz podstawowe kryteria oceny lub wymagania egzaminacyjne</b>	
<ul style="list-style-type: none"> <li>- Wykład z prezentacją multimedialną</li> <li>- ćwiczenia na pracowni komputerowej – rozwiązywanie zadań</li> </ul>		<b>Sposób zaliczenia</b>	
		<ul style="list-style-type: none"> <li>- Zaliczenie na ocenę</li> <li>- Egzamin</li> </ul>	
		<b>Formy zaliczenia</b>	
		<ul style="list-style-type: none"> <li>- egzamin pisemny z pytaniami (zadaniami) otwartymi</li> <li>- egzamin pisemny testowy</li> </ul>	
		<b>Podstawowe kryteria oceny</b>	
		Ocena z ćwiczeń na podstawie wyników z prac domowych oraz aktywności na zajęciach. Ocena z wykładu na podstawie wyników testu sprawdzającego rozumienie fragmentów kodu.	
<b>Sposób weryfikacji założonych efektów uczenia się</b>			

zakładany efekt kształcenia	egzamin	projekt	sprawdzian	obserwacja postawy studenta
<b>Wiedza</b>				
K_W05	x		x	
P_W1	x		x	
P_W2	x		x	
<b>Umiejętności</b>				
K_U06		x	x	
P_U09		x	x	
P_U1		x	x	
P_U2		x	x	
P_U3		x	x	
<b>Kompetencje</b>				
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)
	<b>Kompetencje społeczne (postawy)</b>
	Student: <ul style="list-style-type: none"><li>• P_K1: rozumie potrzebę nauki innych języków programowania (K_K01)</li></ul>
<b>Kontakt</b>	
maciej.dziemianczuk@inf.ug.edu.pl	