



**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		11.3.0804	
<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 Włodzimierz Bzyl; mgr Maciej Dziemiańczuk; Magdalena Zakrzewska; prof. UG, dr hab. Tomasz Dzido; dr Andrzej Borzyszkowski; mgr inż. Anna Nenca; mgr Mateusz Miotk			
<b>Formy zajęć, sposób ich realizacji i przypisana im liczba godzin</b>		<b>Liczba punktów ECTS</b>	
<b>Formy zajęć</b>		10 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>			
Ćw. laboratoryjne: 40 godz., Wykład: 40 godz.			
<b>Cykl dydaktyczny</b>			
2015/2016 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>	
		egzamin pisemny testowy	
		<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 kształcenia</b>			

zakładany efekt kształcenia	egzamin	kolokwium	projekt	referat	raport	aktywność w dyskusji	obserwacja postawy	
Wiedza								
K_W02	x			x			x	
K_W03				x		x	x	
Umiejętności								
K_U04	x					x		
K_U05			x				x	
Kompetencje								
K_K02			x			x	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

**Efekty kształcenia**

**(obszarowe i kierunkowe)**

K\_W02: ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie programowania, algorytmów i złożoności, architektury systemów komputerowych, systemów operacyjnych, technologii sieciowych, języków i paradygmatów programowania, baz danych, inżynierii oprogramowania, języków formalnych, K\_W03: zna podstawowe metody projektowania, analizowania i programowania algorytmów, K\_W04: zna podstawowe konstrukcje programistyczne oraz struktury danych, K\_W08: ma ogólną wiedzę na temat różnych paradygmatów programowania i języków programowania (imperatywny, obiektowy, assembler); szczegółowo zna metody projektowania i programowania obiektowego, K\_W12: zna podstawowe zasady bezpieczeństwa i higieny

**Wiedza**

Odróżnia błędy składniowe w kodzie programów od błędów logicznych. Rozumie krótkie fragmenty kodu. Zna podstawowe konstrukcje języka C. Wie jakie są biblioteki standardowe języka i co zawierają.

**Umiejętności**

Student:

- umie napisać program w C rozwiązujący konkretny problem
- umie zaprogramować w C prosty algorytm
- w swoich programach potrafi skorzystać z gotowych bibliotek
- potrafi uruchamiać programy
- potrafi zaadaptować cudzy kod do swoich potrzeb

**Kompetencje społeczne (postawy)**

Student rozumie potrzebę dalszego kształcenia.

<p>pracy w zawodzie informatyka</p> <p>K_U01 potrafi zastosować wiedzę matematyczną do formułowania, analizowania i rozwiązywania prostych zadań związanych z informatyką, K_U03 potrafi pracować indywidualnie i w zespole informatyków, w tym także potrafi zarządzać swoim czasem oraz podejmować zobowiązania i dotrzymywać terminów, K_U04 potrafi porozumiewać się przy użyciu różnych technik w środowisku zawodowym oraz w innych środowiskach, w tym w języku angielskim oraz z wykorzystaniem narzędzi informatycznych, K_U05 potrafi pisać, uruchamiać i testować programy w wybranym środowisku programistycznym, K_U06 projektuje, analizuje pod kątem poprawności i złożoności obliczeniowej oraz programuje algorytmy; wykorzystuje podstawowe techniki algorytmiczne i struktur danych, K_U08 posługuje się przyjętymi formatami reprezentacji różnego rodzaju danych stosownie do sytuacji, K_U16 ocenia przydatność różnych paradygmatów i związanych z nimi środowisk programistycznych do rozwiązywania różnego typu problemów, K_U17 potrafi ocenić, na podstawowym poziomie, przydatność metod i narzędzi informatycznych, K_U18 potrafi - zgodnie z zadaną specyfikacją - zaprojektować oraz zrealizować prosty system informatyczny, K_U20 jest przygotowany do efektywnego uczestniczenia w inspekcji oprogramowania</p> <p>K_K02 potrafi precyzyjnie formułować pytania, służące pogłębieniu własnego zrozumienia danego tematu lub odnalezieniu brakujących elementów rozumowania</p>	
<p><b>Kontakt</b></p> <p>matwb@ug.edu.pl</p>	