

KARTA KURSU

Nazwa	Programowanie
Nazwa w j. ang.	Programming

Koordinator	dr inż. Magdalena Andrzejewska	Zespół dydaktyczny
Punktacja ECTS*	st. stacjonarne: 7 st. niestacjonarne: 7	mgr Andrzej Borówka mgr inż. Piotr Kukuła dr Zdobysław Świerczyński dr Mariusz Wojciechowski mgr Katarzyna Wójcik

Opis kursu (cele kształcenia)

Celem przedmiotu jest zapoznanie studentów z zasadami programowania imperatywnego o cechach strukturalnych i proceduralnych, metodami projektowania i implementacji nieskomplikowanych programów. Po zakończeniu kursu student będzie znał podstawy programowania w języku C i umiał je stosować. Przedmiot prowadzony jest w języku polskim.

Warunki wstępne

Wiedza	Student zna podstawowe pojęcia związane z programowaniem.
Umiejętności	Student posiada umiejętność opisanie prostych algorytmów.
Kursy	Wstępne kursy nie są wymagane.

Efekty kształcenia

	Efekt kształcenia dla kursu	Odniesienie do efektów kierunkowych
Wiedza	Po zakończeniu kursu student:	
	W01: zna podstawy dobrego programowania, w tym, rozumie ideę programowania strukturalnego i proceduralnego w języku programowania wysokiego poziomu.	K_W03
	W02: posiada podstawową wiedzę z zakresu składni i semantyki języka C, w tym:	K_W03
	<ul style="list-style-type: none"> - zna zasady konstruowania programu w języku C, - potrafi podać ogólną klasyfikację struktur danych, - rozumie pojęcie wskaźnika oraz rekurencji, - zna wybrane funkcje z biblioteki standardowej języka C, w szczególności wybrane: funkcje wejścia/wyjścia, funkcje operujące na ciągach znakowych oraz funkcje dynamicznej alokacji pamięci. 	

	Efekt kształcenia dla kursu	Odniesienie do efektów kierunkowych
Umiejętności	Po zakończeniu kursu student: U01: samodzielnie pisze proste programy w języku C, w których -poprawnie używa standardowych funkcji wejścia/wyjścia, -poprawnie wykorzystuje podstawowe instrukcje sterujące: warunkowe i iteracyjne, -poprawnie definiuje i wykorzystuje proste i złożone struktury danych takie jak: tablice jednowymiarowe, tablice dwuwymiarowe i struktury, -poprawnie deklaruje, definiuje i wywołuje funkcje, w tym również funkcje rekurencyjne oraz z argumentami, którymi są wskaźniki, tablice jednowymiarowe, tablice dwuwymiarowe i struktury.	K_U04
	U02: umie kompilować i uruchamiać programy w języku C oraz śledzić krokowo ich wykonywanie, analizować i poprawiać błędy składniowe i błędy wykonania.	K_U04

	Efekt kształcenia dla kursu	Odniesienie do efektów kierunkowych
Kompetencje społeczne	Po zakończeniu kursu student: K01: potrafi korzystać z różnych źródeł informacji (w tym zasobów sieci Internet) do poszerzania własnej wiedzy i zdobywania nowych umiejętności z zakresu programowania.	K_K02
	K02: potrafi przekazywać wiedzę informatyczną w sposób zrozumiały dla innych.	K_K01

Studia stacjonarne

Organizacja											
Forma zajęć	Wykład (W)	Ćwiczenia w grupach									
		A		K		L		S		P	E
Liczba godzin	30					60					

Studia niestacjonarne

Organizacja											
Forma zajęć	Wykład (W)	Ćwiczenia w grupach									
		A		K		L		S		P	E
Liczba godzin	20					40					

Opis metod prowadzenia zajęć

Podczas zajęć laboratoryjnych studenci rozwiązują zadania, analizują przykładowe programy oraz prezentują i omawiają rozwiązania zadań.

Formy sprawdzania efektów kształcenia

	E – learning	Gry dydaktyczne	Ćwiczenia w szkole	Zajęcia terenowe	Praca laboratoryjna	Projekt indywidualny	Projekt grupowy	Udział w dyskusji	Referat	Praca pisemna (esej)	Egzamin ustny	Egzamin pisemny	Inne
W01					X			X				X	
W02					X			X				X	
U01					X			X				X	
U02					X			X					
K01								X					
K02								X					

Kryteria oceny	<p>Osiągnięcie efektów kształcenia podanych powyżej uprawnia studentów do uzyskania oceny nie wyższej niż dostateczna. Ocenę dobrą lub bardzo dobrą może uzyskać student, który:</p> <ul style="list-style-type: none"> - wykazuje się umiejętnością efektywnej realizacji bardziej złożonych programów (zagnieżdżone instrukcje warunkowe, zagnieżdżone pętle), - prawidłowo dobiera poznane struktury danych do rozwiązywanego problemu, - wyznacza wartości złożonych wyrażeń na podstawie priorytetów operatorów, - rozumie pojęcia czasu życia i zasięgu zmiennych, - rozumie i stosuje zaawansowane wskaźniki, w tym wskaźniki do funkcji.
----------------	---

Uwagi	
-------	--

Treści merytoryczne (wykaz tematów)

<ol style="list-style-type: none"> 1. Język C. Podstawowe typy danych, zmienne i stałe, operatory i wyrażenia, priorytety operatorów. 2. Struktura programu w języku C, komentarze, formatowanie kodu źródłowego, wybrane dyrektywy preprocesora. 3. Obsługa standardowego wejścia/wyjścia. Kompilowanie i śledzenie wykonania programu. 4. Składnia i semantyka instrukcji - instrukcje przypisania, grupujące, sterujące - warunkowe i iteracyjne. 5. Funkcje – prototyp, definiowanie, wywołanie, parametry (formalne, aktualne) i zmienne lokalne funkcji, mechanizm przekazywania parametrów. 6. Tablice jednowymiarowe, tablice dwuwymiarowe. Funkcje korzystające z tablic - argumenty tablicowe. 7. Typ wskaźnikowy - pojęcie wskaźnika, funkcje wykorzystujące notację wskaźnikową, związek wskaźników i tablic, działania na wskaźnikach. Zaawansowane wykorzystanie wskaźników, dynamiczny przydział pamięci. Wskaźniki do funkcji. 8. Podstawy pracy z ciągami tekstowymi. 9. Struktury - wskaźniki do struktur, tablice struktur, funkcje a struktury. 10. Rekurencja.

Wykaz literatury podstawowej

Wskazane przez prowadzącego rozdziały:
<ol style="list-style-type: none"> 1. S. Prata, <i>Język C. Szkoła programowania</i>. Wydanie VI, Helion, 2016 2. P. J. Deitel, H. Deitel, <i>Język C. Solidna wiedza w praktyce</i>. Wydanie VIII, Helion, 2020; 3. R. C. Seacord, <i>Efektywny C. Wprowadzenie do profesjonalnego programowania</i>, Helion 2022 4. W. Malina, P. Mironowicz, <i>Programowanie strukturalne</i>, PWN, 2018.

Wykaz literatury uzupełniającej

1. Loy M., *Język C dla małych urządzeń. Krótki kod o wielkich możliwościach*, Helion 2022
2. Amini K., *Ekstremalny kod w języku C. Współbieżność i programowanie zorientowane obiektowo*, Helion 2021
3. Kochan S. G., *Język C: kompendium wiedzy*, Helion 2016
4. Crawford T., Prinz P., *Język C w pigułce, Kompletny przewodnik*, Promise 2016
5. Jassem K., Ziemkiewicz A., *Sztuka dobrego programowania*, PWN, 2016
6. King K. N., *Język C. Nowoczesne programowanie. Wydanie II*, Helion 2011
7. Tłuczek M., *Programowanie w języku C. Ćwiczenia praktyczne*. Wydanie II, Helion 2011
8. Tondo C. L., Gimpel S. E., *Język ANSI C. Programowanie. Ćwiczenia. Wydanie II*, Helion 2010
9. Kernighan B.W., Ritchie D.M.: *Język ANSI C. Programowanie*. Wydanie II, Helion 2010

Bilans godzinowy zgodny z CNPS (Całkowity Nakład Pracy Studenta) – studia stacjonarne

Liczba godzin w kontakcie z prowadzącymi	Wykład	30
	Konwersatorium (ćwiczenia, laboratorium itd.)	60
	Pozostałe godziny kontaktu studenta z prowadzącym	15
Liczba godzin pracy studenta bez kontaktu z prowadzącymi	Lektura w ramach przygotowania do zajęć	10
	Realizacja zadań domowych (problemowych) po zapoznaniu się z niezbędną literaturą przedmiotu	20
	Przygotowanie projektu lub prezentacji na podany temat (praca indywidualna lub w grupie)	
	Przygotowanie do egzaminu/zaliczenia	40
Ogółem bilans czasu pracy		175
Liczba punktów ECTS w zależności od przyjętego przelicznika		7

Bilans godzinowy zgodny z CNPS (Całkowity Nakład Pracy Studenta) – studia niestacjonarne

Liczba godzin w kontakcie z prowadzącymi	Wykład	20
	Konwersatorium (ćwiczenia, laboratorium itd.)	40
	Pozostałe godziny kontaktu studenta z prowadzącym	10
Liczba godzin pracy studenta bez kontaktu z prowadzącymi	Lektura w ramach przygotowania do zajęć	20
	Realizacja zadań domowych (problemowych) po zapoznaniu się z niezbędną literaturą przedmiotu	30
	Przygotowanie projektu lub prezentacji na podany temat (praca indywidualna lub w grupie)	
	Przygotowanie do egzaminu/zaliczenia	55
Ogółem bilans czasu pracy		175
Liczba punktów ECTS w zależności od przyjętego przelicznika		7