

KARTA KURSU

Nazwa	Podstawy programowania
Nazwa w j. ang.	Introduction to programming

Koordynator	dr Roman Czapla	Zespół dydaktyczny
		dr inż. Magdalena Andrzejewska dr Roman Czapla dr Beata Krzaczek dr Zdobysław Świerczyński mgr Katarzyna Wójcik
Punktacja ECTS*	6	

Opis kursu (cele kształcenia)

Celem przedmiotu jest zapoznanie studentów z zasadami programowania imperatywnego o cechach strukturalnych i proceduralnych, metodami projektowania i zapisu algorytmów oraz 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 uczenia się

	Efekt uczenia się 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_W07
	W02: wie na czym polega działanie translatora i kompilatora języka.	K_W07
	W03: zna wybrane klasyczne algorytmy (w tym wybrane algorytmy wyszukiwania i porządkowania).	K_W07
	W04: posiada podstawową wiedzę z zakresu składni, semantyki i konstrukcji języka C, w tym -potrafi podać ogólną klasyfikację struktur danych, wskazuje różnice pomiędzy typem prostym i złożonym oraz zna sposoby reprezentacji danych w pamięci komputera, -rozumie pojęcie wskaźnika.	K_W07

	Efekt uczenia się dla kursu	Odniesienie do efektów kierunkowych
Umiejętności	Po zakończeniu kursu student: U01: projektuje i zapisuje w postaci schematów blokowych oraz pseudokodu proste algorytmy, analizuje ich działanie oraz umie wykorzystać klasyczne algorytmy w sytuacjach problemowych.	K_U05 K_U02
	U02: samodzielnie pisze proste programy w języku C, w których -poprawnie definiuje i wykorzystuje proste i złożone struktury danych takie jak: tablice jednowymiarowe, tablice dwuwymiarowe, struktury, -poprawnie deklaruje, definiuje i wywołuje funkcje, w tym również z argumentami, którymi są wskaźniki, tablice oraz struktury.	K_U05
	U03: umie kompilować i uruchamiać programy w języku C oraz śledzić krokowo ich wykonywanie, analizować i poprawiać błędy składniowe i błędy wykonania.	K_U05

	Efekt uczenia się 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_K01
	K02: potrafi przekazywać wiedzę informatyczną w sposób zrozumiały dla innych.	K_K03

Studia stacjonarne

		Organizacja										
Forma zajęć	Wykład (W)	Ćwiczenia w grupach										
		A		K		L		S		P		E
Liczba godzin	20					45						

Studia niestacjonarne

		Organizacja										
Forma zajęć	Wykład (W)	Ćwiczenia w grupach										
		A		K		L		S		P		E
Liczba godzin	15					30						

Opis metod prowadzenia zajęć

Podczas pracy laboratoryjnej studenci rozwiązują zadania zdefiniowane przez prowadzącego zajęcia. Na ćwiczeniach na bieżąco omawiane (dyskutowane) są zagadnienia poruszane podczas wykładów.

Formy sprawdzania efektów uczenia się

	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	
W03					X			X				X	
W04					X			X				X	
U01					X			X				X	
U02					X			X				X	
U03					X			X					
K01								X					
K02								X					

Kryteria oceny

Osiągnięcie efektów kształcenia podanych powyżej uprawnia studentów do uzyskania oceny nie wyższej niż dostateczna. Ocenę dobrą lub bardzo dobrą może uzyskać student, który:

- wykazuje się umiejętnością konstruowania algorytmów o większej złożoności (zagnieżdżone instrukcje warunkowe, zagnieżdżone pętle) oraz ich efektywną realizacją w języku programowania C,
- prawidłowo dobiera poznane struktury danych do rozwiązywanego problemu,
- wyjaśnia mechanizm przekazywania parametrów do funkcji.

Uwagi

Treści merytoryczne (wykaz tematów)

1. Wprowadzenie do algorytmiki i programowania - pojęcie algorytmu i programu komputerowego, sposoby zapisu i reprezentacji algorytmów, charakterystyka języków programowania (w tym wysokiego poziomu), proces kompilacji, sposoby reprezentacji liczb w systemie komputerowym.
2. Język C. Podstawowe typy danych, deklaracje – zmienne i stałe, podstawowe (wybrane) operatory i wyrażenia, priorytety operatorów.
3. Struktura programu w języku C, komentarze, formatowanie kodu źródłowego, wybrane dyrektywy preprocesora.
4. Obsługa standardowego wejścia/wyjścia. Kompilowanie i śledzenie wykonania programu.
5. Składnia i semantyka instrukcji- instrukcje przypisania, grupujące, sterujące - warunkowe i iteracyjne.
6. Funkcje – prototyp, definiowanie, wywołanie, parametry (formalne, aktualne) i zmienne lokalne funkcji, mechanizm przekazywania parametrów.
7. Tablice jednowymiarowe, tablice dwuwymiarowe. Funkcje korzystające z tablic - argumenty tablicowe.
8. Typ wskaźnikowy - pojęcie wskaźnika, funkcje wykorzystujące notację wskaźnikową.
9. Struktury- wskaźniki do struktur, tablice struktur, funkcje a struktury.

Wykaz literatury podstawowej

Wskazane przez prowadzącego rozdziały:

1. Deitel Paul. J., Deitel H., *Język C. Solidna wiedza w praktyce. Wydanie VIII*, Helion 2020
2. Malina W., Mironowicz P., *Programowanie strukturalne*, PWN 2018
3. Prata S., *Język C. Szkoła programowania. Wydanie VI*, Helion 2016
4. Crawford T., Prinz P., *Język C w pigułce, Kompletny przewodnik*, Promise 2016

Wykaz literatury uzupełniającej

1. Kochan S. G., *Język C: kompendium wiedzy*, Helion 2016
2. King K. N., *Język C. Nowoczesne programowanie. Wydanie II*, Helion 2011
3. Tłuczek M., *Programowanie w języku C. Ćwiczenia praktyczne. Wydanie II*, Helion 2011
4. Tondo C. L., Gimpel S. E., *Język ANSI C. Programowanie. Ćwiczenia. Wydanie II*, Helion 2010
5. 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	20
	Konwersatorium (ćwiczenia, laboratorium itd.)	45
	Pozostałe godziny kontaktu studenta z prowadzącym	15
Liczba godzin pracy studenta bez kontaktu z prowadzącymi	Lektura w ramach przygotowania do zajęć	30
	Przygotowanie krótkiej pracy pisemnej lub referatu po zapoznaniu się z niezbędną literaturą przedmiotu	
	Przygotowanie projektu lub prezentacji na podany temat (praca w grupie)	
	Przygotowanie do egzaminu/zaliczenia	40
Ogółem bilans czasu pracy		150
Liczba punktów ECTS w zależności od przyjętego przelicznika		6

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

Liczba godzin w kontakcie z prowadzącymi	Wykład	15
	Konwersatorium (ćwiczenia, laboratorium itd.)	30
	Pozostałe godziny kontaktu studenta z prowadzącym	10
Liczba godzin pracy studenta bez kontaktu z prowadzącymi	Lektura w ramach przygotowania do zajęć	45
	Przygotowanie krótkiej pracy pisemnej lub referatu po zapoznaniu się z niezbędną literaturą przedmiotu	
	Przygotowanie projektu lub prezentacji na podany temat (praca w grupie)	
	Przygotowanie do egzaminu/zaliczenia	50
Ogółem bilans czasu pracy		150
Liczba punktów ECTS w zależności od przyjętego przelicznika		6