

KARTA KURSU

Nazwa	Wstęp do programowania		
Nazwa w j. ang.	Introduction to programming		
Kod		Punktacja ECTS*	3
Koordinator	dr Łukasz T. Stępień	Zespół dydaktyczny: dr Łukasz T. Stępień dr Anna Wojciechowska	

Opis kursu (cele kształcenia)

Celem przedmiotu jest zapoznanie studentów z zasadami programowania 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 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_W10
	W02: wie na czym polega działanie kompilatora języka.	K_W10

Studia stacjonarne

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

Studia niestacjonarne

		Organizacja									
Forma zajęć	Wykład (W)	Ćwiczenia w grupach									
		A		K		L		S		P	
Liczba godzin	8					15					

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 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			X
W02					X			X		X			X
W03					X			X		X			X
U01					X			X		X			X
U02					X			X		X			X
K01					X			X		X			X
K02					X			X		X			X

Kryteria oceny	<p>Sposoby sprawdzania osiągnięć edukacyjnych studentów: Dwa kolokwia praktyczne.</p> <p>Do oceny prac praktycznych stosuje się odniesienie wg następującej skali:</p> <ul style="list-style-type: none"> • 51% - 60% ocena 3,0 • 61% - 70% ocena 3,5 • 71% - 80% ocena 4,0
----------------	---

- 81% - 90% ocena 4,5
- 91% - 100% ocena 5,0

Ocena końcowa jest średnią arytmetyczną ocen z dwu kolokwίων.

Przewiduje się jedną poprawę każdego kolokwium.

Kolokwium1

DST:

Student potrafi poprawnie stworzyć programy rozwiązujące proste problemy, wykorzystując odpowiednie struktury danych (m.in. tablice jednowymiarowe) oraz konstrukcje programistyczne dostępne w języku C (wykorzystuje instrukcje sterujące: *if, if-else, switch, while, do-while, for*, zagnieżdżanie instrukcji, instrukcje przerwań *break* i *continue*).

DB:

Student spełnia kryteria uzyskania oceny DST, a także potrafi poprawnie stworzyć programy rozwiązujące średnio zaawansowane problemy wykorzystując odpowiednie struktury danych (m.in. tablice jednowymiarowe) oraz konstrukcje programistyczne dostępne w języku C (wykorzystuje instrukcje sterujące: *if, if-else, switch, while, do-while, for*, zagnieżdżanie instrukcji, instrukcje przerwań *break* i *continue*).

BDB:

Student spełnia kryteria uzyskania oceny DB, a także potrafi poprawnie stworzyć programy rozwiązujące zaawansowane problemy wykorzystując odpowiednie struktury danych (m.in. tablice jednowymiarowe) oraz konstrukcje programistyczne dostępne w języku C (wykorzystuje instrukcje sterujące: *if, if-else, switch, while, do-while, for*, zagnieżdżanie instrukcji, instrukcje przerwań *break* i *continue*).

Kolokwium2

DST:

Student potrafi poprawnie stworzyć programy rozwiązujące proste problemy wykorzystujące proste i złożone struktury danych (tablice dwuwymiarowe), a także programy zawierające zdefiniowane przez studenta funkcje, w tym również z parametrami, którymi są wskaźniki, tablice oraz struktury. Student potrafi też poprawnie stworzyć programy rozwiązujące proste problemy, przy wykorzystaniu technik rekurencyjnych.

DB:

Student spełnia kryteria uzyskania oceny DST, a także potrafi poprawnie stworzyć programy rozwiązujące średnio zaawansowane problemy wykorzystujące proste i złożone struktury danych takie jak: tablice jednowymiarowe, tablice dwuwymiarowe, struktury, a także programy zawierające zdefiniowane przez studenta funkcje, w tym również z parametrami, którymi są wskaźniki, tablice oraz struktury. Student potrafi też poprawnie stworzyć programy rozwiązujące średnio zaawansowane problemy, przy wykorzystaniu technik rekurencyjnych.

BDB:

Student spełnia kryteria uzyskania oceny DB, a także potrafi poprawnie stworzyć programy rozwiązujące zaawansowane problemy wykorzystujące proste i złożone

struktury danych takie jak: tablice jednowymiarowe, tablice dwuwymiarowe, struktury, a także programy zawierające zdefiniowane przez studenta funkcje, w tym również z parametrami, którymi są wskaźniki, tablice oraz struktury. Student potrafi też poprawnie stworzyć programy rozwiązujące zaawansowane problemy, przy wykorzystaniu technik rekurencyjnych.

Uwagi

Treści merytoryczne (wykaz tematów)

1. Elementy algorytmiki, programowania i tworzenia oprogramowania- pojęcie algorytmu i programu komputerowego, charakterystyka języków programowania (w tym wysokiego poziomu), proces kompilacji i łączenia.
2. Struktura programu w języku C, komentarze, formatowanie kodu źródłowego, wybrane dyrektywy preprocesora.
3. Język C. Podstawowe typy danych, deklaracje – zmienne i stałe, inicjalizacja zmiennych i stałych, tablice jednowymiarowe, tablice dwuwymiarowe. Podstawowe (wybrane) operatory i wyrażenia, priorytety operatorów.
4. Obsługa standardowego wejścia/wyjścia. Kompilowanie i śledzenie wykonania programu.
5. Składnia i semantyka instrukcji- instrukcje przypisania, sterujące - warunkowe i iteracyjne.
6. Funkcje – prototyp, deklaracja, definiowanie, wywołanie, parametry (formalne, aktualne) i zmienne lokalne funkcji, mechanizm przekazywania parametrów.
7. Funkcje korzystające z tablic - argumenty tablicowe.
8. Typ wskaźnikowy - pojęcie wskaźnika, dualizm: nazwa tablicy, jako wskaźnik stały, funkcje wykorzystujące notację wskaźnikową.
9. Rekurencja oraz odróżnienie jej od iteracji

Wykaz literatury podstawowej

Wskazane przez prowadzącego rozdziały:

1. Brian W. Kernighan, Dennis M. Ritchie, „Język ANSI C. Programowanie”. Wydanie II, Helion 2010
2. Brian W. Kernighan, Rob Pike, „Lekcja programowania”, z cyklu „Inżynieria oprogramowania”, WNT, Warszawa 2002
3. Steve Oualline, „Język C. Programowanie”, Helion 2003

Wykaz literatury uzupełniającej

1. Frantisek Franek, „Pamięć w językach C i C++”, RM 2005
2. Kenneth A. Reek, „Język C. Wskaźniki. Vademecum profesjonalisty”, Helion 2003
3. Wiesław Rychlicki, „Od matematyki do programowania”, Helion 2011
4. Victor Shtern, „CORE C++. Inżynieria programowania”, Helion 2003
5. C. L. Tondo, S. E. Gimpel, Język ANSI C. Programowanie. Ćwiczenia. Wydanie II, Helion 2010

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

Liczba godzin w kontakcie z prowadzącymi	Wykład	15
	Konwersatorium (ćwiczenia, laboratorium itd.)	30
	Pozostałe godziny kontaktu studenta z prowadzącym	5
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	10
	Przygotowanie projektu lub prezentacji na podany temat (praca w grupie)	
	Przygotowanie do egzaminu	
Ogółem bilans czasu pracy		90
Liczba punktów ECTS w zależności od przyjętego przelicznika		3

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

Liczba godzin w kontakcie z prowadzącymi	Wykład	8
	Konwersatorium (ćwiczenia, laboratorium itd.)	15
	Pozostałe godziny kontaktu studenta z prowadzącym	15
Liczba godzin pracy studenta bez kontaktu z prowadzącymi	Lektura w ramach przygotowania do zajęć	40
	Przygotowanie krótkiej pracy pisemnej lub referatu po zapoznaniu się z niezbędną literaturą przedmiotu	12
	Przygotowanie projektu lub prezentacji na podany temat (praca w grupie)	
	Przygotowanie do egzaminu	
Ogółem bilans czasu pracy		90
Liczba punktów ECTS w zależności od przyjętego przelicznika		3