KARTA KURSU

Nazwa	Wprowadzenie do programowania
Nazwa w j. ang.	Introduction to Programming

Koordynator	dr Andrzej Serafin	Zespół dydaktyczny
Semestr studiów	I	dr Andrzej Serafin
Punktacja ECTS*	3	

Opis kursu (cele kształcenia)

Celem kursu jest zapoznanie studentów z podstawami modelowania systemów informatycznych za pomocą technik programowania funkcjonalnego i obiektowego przy użyciu języka JavaScript (dialekt Source). Przedmiot prowadzony jest w języku polskim z użyciem terminologii angielskiej.

Warunki wstępne

Wiedza	Student zna podstawy programowania na poziomie szkoły średniej.
Umiejętności	
Kursy	

Efekty uczenia się

	Efekt uczenia się dla kursu	Odniesienie do efektów kierunkowych
Wiedza	W01 – Zna w zaawansowanym stopniu terminologię informatyczną w zakresie programowania. W02 – Posiada podstawową wiedzę dotyczącą programowania.	K_W02 K_W03

Umiejętności	Efekt uczenia się dla kursu	Odniesienie do efektów kierunkowych

	U01 – Posługuje się językiem specjalistycznym z zakresu programowania. Potrafi formułować i analizować problemy programistyczne oraz dobiera możliwe optymalne metody ich rozwiązywania. U02 – Potrafi posługiwać się narzędziami programistycznymi.	K_U01 K_U03

Kompetencje społeczne	Efekt uczenia się dla kursu	Odniesienie do efektów kierunkowych
	K01 – Samodzielnie i odpowiedzialnie podejmuje zadania w zakresie programowania.	K_K02

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

Opis metod prowadzenia zajęć

Wykład teoretyczny z podstaw programowania równolegle z praktycznymi ćwiczeniami programistycznymi. Kurs wykorzystuje metodę nauczania podstaw programowania “Structure and Interpretation of Computer Programs”, opracowaną w latach 80-tych w MIT przez Harolda Abelsona i Gerarda Sussmana w oparciu o język Scheme, zaadaptowaną w 2022 r. do języka JavaScript, łączącą wykład podstaw algorytmiki, teorii systemów i programowania funkcyjnego.

Formy sprawdzania efektów uczenia się

	E–learning	Grydydaktyczn	Ćwiczeniawszk	Zajęciaterenо	Pracalaboratо	Projektindywi	Projektgrupow	Udziałwdyskus	Referat	Pracapisemna(e	Egzaminustny	Egzaminpisemn	Inne

		e	o l e	w e	r y j n a	d u a l n y	y	j i		s ej)		y	
W01					X							X	
W02					X							X	
U01					X							X	
U02					X							X	
K01					X							X	

Kryteria oceny	Obecność na zajęciach, wykonywanie zadań programistycznych podczas ćwiczeń, kolokwium sprawdzające wiedzę teoretyczną.

Treści merytoryczne (wykaz tematów)

1. Podstawy informatyki
1.1. Architektura von Neumanna i maszyna Turinga
1.2. Paradygmat funkcyjny i rachunek Lambda
2. Funkcje
2.1. Elementy programowania funkcyjnego
2.2. Funkcje i procesy
3.3. Funkcyjne modelowanie abstrakcji
3. Dane
3.1. Abstrakcja danych
3.2. Hierarchiczność danych
3.3. Symboliczność danych
3.4. Reprezentacja danych
3.5. Generyczność danych
4. Modelowanie systemów
4.1. Obiekty i stany
4.2. Środowisko
4.3. Zmienność obiektów
4.4. Synchroniczność i współbieżność
4.5. Paradygmat strumieniowy

Wykaz literatury podstawowej

H. Abelson, G. Sussman, Structure and Interpretation of Computer Programs: JavaScript Edition, 2022 (wyd. pol. Struktura i interpretacja programów komputerowych, 2002)

Wykaz literatury uzupełniającej

D. Knuth, Algorithms, Programs, and Computer Science, 1966/1992
M. Minsky, Form and content in computer science, 1970
E. W. Dijkstra, A Short Introduction to the Art of Programming, 1974
J. Backus, Can programming be liberated from the von Neumann style?, 1978
D. Harel, Algorithmics: The Spirit of Computing, 2004 (wyd. pol. Rzecz o istocie informatyki, 2000)
Ch. Petzold, Annotated Turing, 2008
J. Backfield, Becoming Functional, 2014 (wyd. pol. Programowanie funkcyjne, 2015)
R. W. Sebesta, Concepts of Programming Languages, 2019
J. G. Brookshear, Computer Science: An Overview, 2020 (wyd. pol. Informatyka w ogólnym zarysie, 2003)

Bilans godzinowy zgodny z CNPS (Całkowity Nakład Pracy Studenta)

Ilość godzin w kontakcie z prowadzącymi	Wykład	15
	Konwersatorium (ćwiczenia, laboratorium itd.)	15
	Pozostałe godziny kontaktu studenta z prowadzącym	15
Ilość godzin pracy studenta bez kontaktu z prowadzącymi	Lektura w ramach przygotowania do zajęć	15
	Przygotowanie krótkiej pracy pisemnej lub referatu po zapoznaniu się z niezbędną literaturą przedmiotu	0
	Przygotowanie projektu lub prezentacji na podany temat (praca w grupie)	0
	Przygotowanie do egzaminu	15
Ogółem bilans czasu pracy		75
Ilość punktów ECTS w zależności od przyjętego przelicznika		3