

SYLABUS
DOTYCZY CYKLU KSZTAŁCENIA 2023-2027
Rok akademicki 2025/2026

1. PODSTAWOWE INFORMACJE O PRZEDMIOCIE

Nazwa przedmiotu	<i>programowanie funkcyjne</i>
Kod przedmiotu	
Nazwa jednostki prowadzącej kierunek	<i>Instytut Informatyki, Kolegium Nauk Przyrodniczych</i>
Nazwa jednostki realizującej przedmiot	<i>Instytut Informatyki, Kolegium Nauk Przyrodniczych</i>
Kierunek studiów	<i>informatyka</i>
Poziom studiów	<i>studia inżynierskie I-go stopnia</i>
Profil	<i>ogólnoakademicki</i>
Forma studiów	<i>stacjonarne</i>
Rok i semestr/y studiów	<i>rok III semestr 6</i>
Rodzaj przedmiotu	<i>przedmiot kierunkowy</i>
Język wykładowy	<i>język polski, język angielski</i>
Koordynator	<i>dr Krzysztof Balicki</i>
Imię i nazwisko osoby prowadzącej / osób prowadzących	<i>dr Krzysztof Balicki, dr inż. Wojciech Kozioł</i>

1.1 Formy zajęć dydaktycznych, wymiar godzin i punktów ECTS

Semestr (nr)	Wykł.	Ćw.	Konw.	Lab.	Sem.	ZP	Prakt.	Inne (jakie?)	Liczba pkt. ECTS
6	12			13					2

1.2 Sposób realizacji zajęć

zajęcia w formie tradycyjnej

1.3 Forma zaliczenia przedmiotu (z toku)

zaliczenie z oceną

2. WYMAGANIA WSTĘPNE

Podstawy programowania w języku C, Algorytmy i struktury danych, Programowanie obiektowe
--

3. CELE, EFEKTY UCZENIA SIĘ, TREŚCI PROGRAMOWE I STOSOWANE METODY DYDAKTYCZNE**3.1 Cele przedmiotu**

C ₁	Zapoznanie studentów z paradygmatem programowania funkcyjnego jako reprezentanta programowania deklaratywnego i językiem Haskell
C ₂	Wykształcenie u studentów praktycznego stosowania poznanego języka.

3.2 Efekty uczenia się dla przedmiotu

EK (efekt uczenia się)	Treść efektu uczenia się zdefiniowanego dla przedmiotu	Odniesienie do efektów kierunkowych
EK_01	Student zna zasady formułowania i algorytmizacji zadań oraz podstawowe notacje zapisu algorytmów.	K_Wo6
EK_02	Student zna paradygmat programowania funkcyjnego oraz co najmniej jeden język go reprezentujący w stopniu umożliwiającym mu pisanie prostych programów użytkowych.	K_Wo6
EK_03	Student umie ułożyć i analizować (w tym śledzić) algorytm zgodny ze specyfikacją i zapisać go w języku programowania funkcyjnego.	K_Uo2
EK_04	Student potrafi pisać proste programy użytkowe w języku programowania funkcyjnego.	K_Uo2

3.3 Treści programowe**A. Problematyka wykładu**

Przegląd najważniejszych paradygmatów programowania.
Programowanie funkcyjne w języku Haskell; historia i składania języka Haskell; dostępne kompilatory
Porównanie paradygmatu programowania funkcyjnego i paradygmatu programowania imperatywnego na przykładach implementacji wybranych problemów algorytmicznych.

B. Problematyka ćwiczeń laboratoryjnych

Środowisko pracy. Kompilatory i interpretery.
Typy, operatory, funkcje w Haskellu, operacje I/O
Rekurencja, sterowanie, funkcje wyższego rzędu
Implementacja filtrowania słownikowego
Implementacja algorytmu wyszukiwania wszcz

3.4 Metody dydaktyczne

Wykład: wykład problemowy, wykład z prezentacją multimedialną
 Laboratoria: rozwiązywanie zadań, dyskusja

4. METODY I KRYTERIA OCENY**4.1 Sposoby weryfikacji efektów uczenia się**

Symbol efektu	Metody oceny efektów uczenia się (np.: kolokwium, egzamin ustny, egzamin pisemny, projekt, sprawozdanie, obserwacja w trakcie zajęć)	Forma zajęć dydaktycznych (w, ćw, ...)
EK_01	obserwacja w trakcie zajęć	lab
EK_02	kolokwium	lab
EK_03	kolokwium	lab
EK_04	kolokwium	lab

4.2 Warunki zaliczenia przedmiotu (kryteria oceniania)**Wykład**

Zaliczenie bez oceny, efekty kierunkowe EK_01, EK_02 właściwe dla wykładu weryfikowane są w trakcie zajęć laboratoryjnych przy pomocy kolokwium i obserwacji na zajęciach.

Laboratorium

Warunkiem zaliczenia laboratorium jest zaliczenie kolokwium z języka Haskell oraz z języka Prolog. Ocena końcowa jest średnią ocen z tych dwóch kolokwium. Aby zaliczyć kolokwium należy zdobyć przynajmniej połowę maksymalnej liczby punktów. Oceny z kolokwium przyznawane są proporcjonalnie do liczby zdobytych punktów. Pod uwagę brana jest również aktywność na zajęciach, która może obniżyć lub podwyższyć ocenę końcową o pół stopnia.

5. CAŁKOWITY NAKŁAD PRACY STUDENTA POTRZEBNY DO OSIĄGNIĘCIA ZAŁOŻONYCH EFEKTÓW W GODZINACH ORAZ PUNKTACH ECTS

Forma aktywności	Średnia liczba godzin na zrealizowanie aktywności
Godziny kontaktowe wynikające z harmonogramu studiów	25
Inne z udziałem nauczyciela akademickiego (udział w konsultacjach, egzaminie)	3
Godziny niekontaktowe – praca własna studenta (przygotowanie do zajęć, egzaminu, napisanie referatu itp.)	25
SUMA GODZIN	53
SUMARYCZNA LICZBA PUNKTÓW ECTS	2

6. PRAKTYKI ZAWODOWE W RAMACH PRZEDMIOTU

wymiar godzinowy	-
zasady i formy odbywania praktyk	-

7. LITERATURA

Literatura podstawowa:

1. P. Hudak, J. Peterson, J. Fasel: *A Gentle Introduction to Haskell*
<https://www.haskell.org/tutorial/index.html>
2. Bryan O'Sullivan, Don Stewart, John Goerzen: *Real World Haskell*. O'Reilly Media 2008
3. Kees Doets, Jan van Eijck: *The Haskell Road to Logic, Math and Programming*. King's College Publications 2004
4. Graham Hutton: *Programming in Haskell*. Cambridge University Press 2007

Literatura uzupełniająca:

1. Joeren Fokker: *Functional Programming*. Department of Computer Science, Utrecht University 1995 (plik pdf dostępny w Internecie)
2. Hal Daume III, et. al.: *Yet Another Haskell Tutorial*. 2004 (plik pdf dostępny w Internecie)
3. Brian W. Kernighan, Dennis Ritchie: *Język ANSI C*. WNT, Warszawa 2003
4. Bruce Eckel: *Thinking in Java*. Edycja polska, Wydanie 4, Helion, Gliwice, 2006