

1.	Field of study	Mathematics
2.	Faculty	Faculty of Science and Technology
3.	Academic year of entry	2022/2023 (winter term)
4.	Level of qualifications/degree	first-cycle studies
5.	Degree profile	general academic
6.	Mode of study	full-time

Module: Algorithms and Programming

Module code: W4-MT-S1-21-AiP

1. Number of the ECTS credits: 7

2. Learning outcomes of the module			
code	description	learning outcomes of the programme	level of competence (scale 1-5)
AiP_1	testuje na komputerze swoje programy pod względem zgodności z przyjętymi założeniami i ewentualnie je poprawia, objaśnia przebieg działania programów	KN_I_U04 K_U26	5 5
AiP_10	posiada umiejętność oceny ograniczeń narzędzi komputerowych	K_W08	5
AiP_2	formułuje problem w postaci specyfikacji (czyli opisuje dane i wyniki) i wyróżnia kroki w algorytmicznym rozwiązywaniu problemów; zna pojęcie algorytmu i stosuje różne sposoby przedstawiania algorytmów, w tym w języku naturalnym, w postaci schematów blokowych, listy kroków, w pseudokodzie oraz w wybranym języku programowania	KN_I_U03 KN_I_U04 KN_I_W03 K_U25 K_U26 K_W08	5 5 5 5 5 5
AiP_3	zna i zapisuje klasyczne algorytmy za pomocą listy kroków, schematu blokowego lub pseudokodu oraz implementuje je w wybranym języku programowania; zna i omawia sytuacje, w których wykorzystuje się klasyczne algorytmy	KN_I_U03 KN_I_U04 KN_I_W03 K_W08	5 5 5 5
AiP_4	zna podstawowe własności algorytmów; prezentuje przykłady zastosowań algorytmiki w innych dziedzinach nauki	KN_I_U03 KN_I_W03 K_W08	3 3 3
AiP_5	rozwija znajomość algorytmów i wykonuje eksperymenty z algorytmami; rozumie potrzebę programowania z użyciem zaawansowanych algorytmów	KN_I_K02	3

		KN_I_U03 KN_I_W03 K_U25 K_U27	3 3 3 3
AiP_6	zna i rozumie pojęcie złożoności obliczeniowej (czasowej i pamięciowej) oraz notacji asymptotycznej	KN_I_U03 KN_I_W03 K_W08	4 4 4
AiP_7	zapisuje wybrane algorytmy klasyczne w postaci iteracyjnej lub/oraz rekurencyjnej	KN_I_U03 KN_I_U04 KN_I_W03 K_U26	5 5 5 5
AiP_8	porównuje działanie różnych algorytmów dla wybranego problemu, analizuje algorytmy na podstawie ich gotowych implementacji	KN_I_U03 KN_I_W03 K_U26 K_U27	4 4 4 4
AiP_9	zna różne metody i techniki programowania: podejście zachłanne, programowanie dynamiczne, programowanie z nawrotami	KN_I_U03 KN_I_W03 K_W08	3 3 3

3. Module description

Description	1. Elementy algorytmiki: problem i jego specyfikacja; algorytm i różne sposoby jego zapisu. 2. Elementy analizy algorytmów. Rozmiar danych, złożoność obliczeniowa (czasowa i pamięciowa). Typy złożoności: pesymistyczna, optymistyczna, średnia. Notacja asymptotyczna, rzędy wielkości funkcji. 3. Algorytmy iteracyjne i rekurencyjne; metoda dziel i zwyciężaj. 4. Algorytmy klasyczne w tym m.in.: - obliczania pochodnej wielomianu za pomocą schematu Hornera, - algorytmy Euklidesa w wersji iteracyjnej i rekurencyjnej wraz z zastosowaniami, - operujące na liczbach (badania pierwszości liczby, zamiany reprezentacji liczb między pozycyjnymi systemami liczbowymi, działań na ułamkach z wykorzystaniem NWD i NWW), - operujące na tekstach (porównywanie tekstów, wyszukiwania wzorca w tekście metodą naiwną, szyfrowania tekstu metodą Cezara i przestawieniową), - wyszukiwania elementów w dowolnej tablicy (algorytm sekwencyjny) oraz w tablicy uporządkowanej (metoda wyszukiwania binarnego) - sortujące (sortowanie przez wstawianie, przez wybieranie, bąbelkowe, przez scalanie, szybkie), - znajdowania określonego elementu w zbiorze: maksymalnego, lidera oraz idola, - generowania liczb pierwszych metodą sita Eratostenesa, - jednoczesnego wyszukiwania elementu najmniejszego i największego (algorytm iteracyjny oraz rekurencyjny wykorzystujący metodę dziel i zwyciężaj), - szybkiego potęgowania liczb w wersji iteracyjnej i rekurencyjnej, - problem wież z Hanoi (rozwiązanie rekurencyjne). 5. Różne metody i techniki programowania: - podejście zachłanne (wydawania reszty najmniejszą liczbą nominałów, pakowanie plecaka),
--------------------	---

	- programowanie dynamiczne (pakowanie plecaka, szukania najdłuższego wspólnego podciągu), - algorytmy z nawrotami (problem n-hetmanów). 6. Implementacja poznanych algorytmów w wybranym języku programowania wysokiego poziomu.
Prerequisites	Brak

4. Assessment of the learning outcomes of the module			
code	type	description	learning outcomes of the module
AiP_w_1	kolokwium na konwersatorium	Kolokwium pisemne na ostatnich lub przedostatnich zajęciach; zadania podobnego typu do zadań rozwiązywanych w trakcie zajęć konwersatoryjnych	AiP_2, AiP_3, AiP_6, AiP_7, AiP_8, AiP_9
AiP_w_2	kolokwia na laboratorium	Dwa kolokwia w semestrze; zadania podobnego typu do zadań rozwiązywanych w trakcie zajęć laboratoryjnych	AiP_1, AiP_3, AiP_7, AiP_8
AiP_w_3	zadania domowe	ocena zadań domowych; możliwość odpytania z wybranych zagadnień/zadań zadanych na pracę w domu	AiP_1, AiP_10, AiP_2, AiP_3, AiP_4, AiP_5, AiP_6, AiP_7, AiP_8, AiP_9
AiP_w_4	egzamin	Warunkiem przystąpienia do egzaminu jest zaliczenie konwersatorium oraz laboratorium; weryfikacja znajomości pojęć i faktów w oparciu o analizę odpowiedzi na pytania egzaminacyjne	AiP_3, AiP_6, AiP_7, AiP_8, AiP_9

5. Forms of teaching						
code	form of teaching			required hours of student's own work		assessment of the learning outcomes of the module
	type	description (including teaching methods)	number of hours	description	number of hours	
AiP_fs_1	lecture	wykład z wykorzystaniem pomocy audiowizualnych	25	przyswojenie wiadomości z wykładu przy pomocy udostępnionych materiałów wykładowych; lektura uzupełniająca podręczników;	25	AiP_w_4
AiP_fs_2	laboratory classes	praca w laboratorium z wykorzystaniem komputera w oparciu o otwarte środowiska programistyczne	20	praca własna z wykorzystaniem ogólnodostępnego oprogramowania, doskonalenie umiejętności zdobytych podczas zajęć	50	AiP_w_2, AiP_w_3
AiP_fs_3	discussion classes	konwersatorium, w trakcie którego studenci rozwiązują, pod kierunkiem prowadzącego, zadania kształtujące umiejętności wymienione w zestawie efektów kształcenia modułu	15	przyswojenie wiedzy z wykładów, samodzielna praca ze zbiorami zadań,	40	AiP_w_1, AiP_w_3