SYLABUS

DOTYCZY CYKLU KSZTAŁCENIA *2026-2029*

(skrajne daty)

Rok akademicki 2026/2027

1. PODSTAWOWE INFORMACJE O PRZEDMIOCIE

Nazwa przedmiotu	Podstawy programowania
Kod przedmiotu*	
Nazwa jednostki prowadzącej kierunek	Wydział Nauk Ścisłych i Technicznych
Nazwa jednostki realizującej przedmiot	Wydział Nauk Ścisłych i Technicznych Instytut Informatyki
Kierunek studiów	Matematyka
Poziom studiów	studia I stopnia
Profil	ogólnoakademicki
Forma studiów	stacjonarne
Rok i semestr studiów	rok I, semestr 1
Rodzaj przedmiotu	kierunkowy
Język wykładowy	język polski
Koordynator	dr Paweł Pasteczka
Imię i nazwisko osoby prowadzącej / osób prowadzących	dr Paweł Pasteczka

***** *- zgodnie z ustaleniami w Jednostce*

1.1.Formy zajęć dydaktycznych, wymiar godzin i punktów ECTS

Semestr (nr)	Wykł.	Ćw.	Konw.	Lab.	Sem.	ZP	Prakt.	Inne (jakie?)	**Liczba pkt ECTS**
1	30			30					5

1.2. Sposób realizacji zajęć

☐ zajęcia w formie tradycyjnej

TAK zajęcia realizowane z wykorzystaniem metod i technik kształcenia na odległość

1.3 Forma zaliczenia przedmiotu (z toku) (egzamin, zaliczenie z oceną, zaliczenie bez oceny)

Wykład-zaliczenie,

laboratoria - zaliczenie z oceną

2.WYMAGANIA WSTĘPNE

Wiadomości z matematyki i informatyki z zakresu szkoły średniej.

3. CELE, EFEKTY UCZENIA SIĘ , TREŚCI PROGRAMOWE I STOSOWANE METODY DYDAKTYCZNE

3.1 Cele przedmiotu

C1	Przygotowanie do rozwiązywania problemów (również matematycznych) przy użyciu środków informatyki oraz korzystania z komputerów na zajęciach matematycznych.
C2	Zapoznanie studentów z podstawami arytmetyki maszyn cyfrowych, systemów kodowania oraz reprezentacji danych w programach komputerowych.
C3	Wprowadzenie w zagadnienia algorytmicznego opisu rozwiązywania problemów – poszukiwanie, dostrzeganie i konstruowanie algorytmów.
C4	Zapoznanie studentów z podstawowymi metodami projektowania algorytmów, pisania, uruchamiania i testowania programów. Analiza wyników obliczeń.

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 podstawy technik obliczeniowych i programowania, które mogą być wykorzystane we wspomaganiu pracy matematyka oraz rozumie ograniczenia pojawiające się przy ich wykorzystaniu.	K_W05
EK_02	student potrafi rozpoznać i dokonać specyfikacji problemu, który można rozwiązać algorytmicznie, student dla danego problemu potrafi ułożyć i przeanalizować algorytm zgodny ze specyfikacją i zapisać go w odpowiednim języku programowania, a następnie testować napisany samodzielnie program komputerowy i dokonywać niezbędnych korekt.	K_U14
EK_03	student wyraża własne opinie na temat teoretycznych i praktycznych zagadnień z algorytmiki dotyczących danego problemu oraz formułuje pytania służące zrozumieniu badanego problemu, a w razie potrzeby uzupełnia swoje kompetencje.	K_K01
EK_04	student jest gotów do prezentowania krytycznej postawy wobec odbieranych treści, ma świadomość błędów, które mogą się pojawić przy układaniu algorytmów oraz zapisach w używanych językach programowania.	K_K02, K_K03

3.3 Treści programowe

A. Problematyka wykładu

Treści merytoryczne
Algorytmy: Algorytmy i sposoby ich przedstawiania, konwencja notacyjna, struktury algorytmów (liniowe, z rozgałęzieniami, iteracyjne, rekurencyjne), własności algorytmów (poprawność, złożoność i efektywność), algorytmy rekurencyjne.
Komputery:

Schemat funkcjonalny komputera, organizacja pamięci. Pozycyjne systemy liczbowe, konwersje zapisów liczb, stało- i zmiennopozycyjny sposób zapisu liczb na skończonej liczbie miejsc, załamanie się własności działań arytmetycznych dla liczb reprezentowanych na skończonej liczbie miejsc, kody maszynowe liczb.
Budowa programu na przykładzie języka C++: Związek programu z algorytmem. C++ jako imperatywny język programowania. Stałe i zmienne. Podstawowe instrukcje. Typy danych, literały, operatory i wyrażenia. Operacje wejścia i wyjścia.
Instrukcje złożone: Instrukcje warunkowe. Różne rodzaje pętli. Instrukcja switch. Zagnieżdżanie instrukcji. Instrukcje przerwań break i continue.
Funkcje: Funkcje. Przekazywanie parametrów do funkcji. Rekurencja. Wybrane funkcje biblioteczne.
Zmienne złożone: Typ tablicowy, Typ plikowy. Łańcuchy znaków. Operacje na zmiennych w/w typów.
Wskaźniki i organizacja pamięci: Wskaźniki, dynamiczny przydział pamięci. Stos i sterta.

B. Problematyka laboratoriów

Treści merytoryczne
Algorytmy: Zapis algorytmiczny rozwiązania prostych problemów. Różne sposoby zapisu algorytmów - schematy blokowe. Testowanie algorytmu.
Narzędzia programistyczne: Wprowadzenie do kompilatora, debuggera i narzędzi wspierających programowanie. Analiza rezultatów kompilacji przykładowych programów. Interpretacja rodzajów błędów (błędy kompilatora, linkera).
Budowa programu na przykładzie języka C++: Związek programu z algorytmem. Stałe i zmienne. Podstawowe instrukcje. Tworzenie prostych programów z użyciem operacji arytmetycznych, logicznych i standardowych funkcji wejścia/wyjścia.
Instrukcje złożone: Instrukcje warunkowa if. Użycie pętli do, while i for. Instrukcja switch.
Zmienne złożone: Użycie zmiennych typu tablicowego. Obsługa plików.
Funkcje: Tworzenie prostych funkcji i ich użycie w programie. Używanie zmiennych globalnych i lokalnych. Iteracja. Rekurencja.
Wskaźniki i organizacja pamięci: Wskaźniki, dynamiczny przydział pamięci.

3.4 Metody dydaktyczne

Wykład- Wykład z prezentacją multimedialną.
Laboratoria- rozwiązywanie zadań, dyskusja, projektowanie algorytmów, tworzenie programów.

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	projekt zaliczeniowy	wykład, laboratoria
EK_ 02	projekt zaliczeniowy	wykład, laboratoria
EK_ 03	projekt zaliczeniowy	laboratoria
EK_ 04	projekt zaliczeniowy	laboratoria

4.2 Warunki zaliczenia przedmiotu (kryteria oceniania)

Wykład

Zaliczenie wykładu na podstawie testu

Laboratoria:

Warunkiem uzyskania zaliczenia z laboratoriów jest uzyskanie co najmniej 51 procent punktów możliwych do uzyskania z projektów. W przypadku niespełnienia powyższych warunków student może wysłać projekty w terminie poprawkowym. Aktywność podczas zajęć premiowana jest dodatkowymi punktami. Oceny wystawiane są według następującej skali procentowej:

0	–	50	niedostateczny
51	–	60	dostateczny
61	–	70	plus dostateczny
71	–	80	dobry
81	–	90	plus dobry
91	–	100	bardzo dobry

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

Forma aktywności	Średnia liczba godzin na zrealizowanie aktywności
Godziny z harmonogramu studiów	60
Inne z udziałem nauczyciela (udział w konsultacjach, egzaminie)	5
Godziny niekontaktowe – praca własna studenta (przygotowanie do zajęć, egzaminu, napisanie referatu itp.)	60
SUMA GODZIN	125
SUMARYCZNA LICZBA PUNKTÓW ECTS	5

** Należy uwzględnić, że 1 pkt ECTS odpowiada 25-30 godzin całkowitego nakładu pracy studenta.*

6. PRAKTYKI ZAWODOWE W RAMACH PRZEDMIOTU

wymiar godzinowy	nie dotyczy
zasady i formy odbywania praktyk	nie dotyczy

7. LITERATURA

Literatura podstawowa:

1. D. Harel, Rzecz o istocie informatyki: algorytmika, WNT, Warszawa 2008.

2. N. Wirth, Wstęp do programowania systematycznego, WNT, Warszawa 1987.

3. N. Wirth, Algorytmy+struktury danych=programy, WNT, Warszawa 2004.

4. Język C++ : szkoła programowania / Stephen Prata ; [tł. Przemysław Szeremiota]. - Gliwice : Wydawnictwo Helion, cop. 2013.

5. Strona internetowa www.algorytm.edu.pl

Literatura uzupełniająca:

1. A.V. Aho, J.E. Hopcroft i J.D. Ullman, Algorytmy i struktury danych, Helion, Warszawa 2003

2. L. Banachowski, K. Diks i W. Rytter, Algorytmy i struktury danych, WNT, Warszawa 2006.

3. T.H. Cormen, Ch.E. Leiserson, R.L. Rivest i C. Stein, Wprowadzenie do algorytmów, WNT, Warszawa 2007.

4. M. Sysło, Algorytmy, WSiP, Warszawa 2008.

5. http://wazniak.mimuw.edu.pl

Akceptacja Kierownika Jednostki lub osoby upoważnionej