

# AKADEMIA TECHNICZNO-INFORMATYCZNA W NAUKACH STOSOWANYCH

## KARTA OPISU PRZEDMIOTU

<b>Wydział</b>		<b>Informatyki</b>	
<b>Kierunek</b>		<b>Informatyka</b>	
<b>Specjalność</b>			
<b>Semestr</b>	<b>II</b>	<b>Program studiów, dla którego obowiązuje sylabus</b>	<b>2025/2026</b>
<b>Stopień studiów</b>	<b>I</b>		

<b>Nazwa przedmiotu</b>	<b>C++</b>			
<b>Kod przedmiotu</b>	<b>C++</b>			
<b>Łączna liczba godzin</b>	<b>45</b>	<b>Tryb</b>	<b>stacjonarny</b>	<b>niestacjonarny</b>
<b>Profil kształcenia</b>	<b>Ogólnoakademicki (A)   Praktyczny (P)</b>			
<b>Forma zajęć</b>	<b>wykład + laboratorium</b>			
<b>Język przedmiotu</b>	<b>polski</b>			
<b>Liczba punktów ECTS</b>	<b>5 (3+2)</b>			

<b>Prowadzący zajęcia</b>	
<b>Forma prowadzonych zajęć</b>	<b>Wykład</b>
<b>Wymiar zajęć</b>	<b>27 h</b>
<b>Stopień (tytuł) naukowy</b>	
<b>Imię</b>	
<b>Nazwisko</b>	

<b>Prowadzący zajęcia</b>	
<b>Forma prowadzonych zajęć</b>	<b>Laboratorium</b>
<b>Wymiar zajęć</b>	<b>18 h</b>
<b>Stopień (tytuł) naukowy</b>	
<b>Imię</b>	
<b>Nazwisko</b>	

<b>Wymagania wstępne</b>	Umiejętność obsługi komputera z systemem Windows. Podstawowa wiedza z zakresu matematyki i logiki.
<b>Założenia i cele przedmiotu</b>	Zapoznanie studentów z podstawami i zaawansowanymi aspektami języka C++, w tym programowaniem obiektowym i korzystaniem ze standardowych bibliotek, rozwinięcie umiejętności praktycznych w zakresie pisania, kompilowania, testowania i debugowania programów, kształtowanie kompetencji niezbędnych do samodzielnego tworzenia efektywnych i bezpiecznych aplikacji.
<b>Metody dydaktyczne</b>	Zajęcia prowadzone w formie wykładów oraz laboratoriów, na których studenci zajmują się praktyczną realizacją zadań z zakresu podstawowych zagadnień C++. Do zajęć każdorazowo są przygotowywane scenariusze umożliwiające pracę studenta we własnym tempie, niezależnym od reszty grupy. Praca zaliczeniowa/projekt na koniec semestru.

# AKADEMIA TECHNICZNO-INFORMATYCZNA W NAUKACH STOSOWANYCH

Efekty uczenia się (odniesienie do charakterystyk poziomów Polskiej Ramy Kwalifikacji)		Odniesienie do efektów dla kierunku	Odniesienie do efektów uczenia się wg Polskiej Ramy Kwalifikacji
WIEDZA – absolwent zna i rozumie:	01. Podstawy składni i semantyki języka C++, w tym proces kompilacji i linkowania, oraz różnice między językami proceduralnymi a obiektowymi. 02. Koncepcje programowania obiektowego w C++, w tym tworzenie klas, obiektów, hermetyzację, dziedziczenie i polimorfizm oraz ich wpływ na strukturę i organizację kodu. 03. Podstawowe struktury danych (tablice, wskaźniki, listy, wektory, mapy, sety) oraz związane z nimi algorytmy i złożoności obliczeniowe istotne przy projektowaniu wydajnych aplikacji. 04. Zasady wykorzystania szablonów, iteratorów oraz algorytmów biblioteki STL. 05. Mechanizmy obsługi wyjątków, zarządzania pamięcią (w tym dynamicznej alokacji) oraz znaczenie stosowania dobrych praktyk programistycznych w celu zapewnienia jakości i bezpieczeństwa kodu.	K_W02 K_W06 K_W07	P6S_WG P6S_WG_INŻ
UMIEJĘTNOŚCI – absolwent potrafi:	01. Napisać, skompilować i uruchomić program w C++, korzystając z dostępnych środowisk programistycznych i narzędzi, oraz samodzielnie diagnozować proste problemy związane z kompilacją. 02. Zaprojektować i zaimplementować klasy oraz hierarchie klas, stosując zasady obiektowości (dziedziczenie, polimorfizm), aby stworzyć strukturalnie poprawne, rozszerzalne i czytelne programy. 03. Wykorzystać szablony i kontenery STL do implementacji złożonych struktur danych i algorytmów, integrując je z innymi elementami aplikacji w celu zwiększenia elastyczności kodu. 04. Realizować operacje wejścia/wyjścia na standardowych strumieniach oraz plikach tekstowych i binarnych, zapewniając poprawne przetwarzanie danych w praktycznych projektach. 05. Identyfikować, obsługiwać oraz rozwiązywać problemy związane z błędami w czasie wykonania, wykorzystując mechanizmy wyjątków, narzędzia debuggera oraz testowanie, aby zapewnić wysoką jakość i niezawodność tworzonego oprogramowania.	K_U15	P6S_UW P6S_UW_INŻ
KOMPETENCJE SPOŁECZNE – absolwent jest gotów do	01. Ciągłego doskonalenia się, podnoszenia kompetencji zawodowych, osobistych i społecznych. 02. Pracy w zespole i przyjmowania w nim różnych ról.	K_K03 K_K04	P6S_UU P6S_UO P6S_KR

## AKADEMIA TECHNICZNO-INFORMATYCZNA W NAUKACH STOSOWANYCH

Treści programowe		
Lp.	Tematyka zajęć	Liczba godzin
<b>Forma zajęć – wykład</b>		
1	Wprowadzenie do C++: środowisko, kompilacja, linkowanie, składnia, proste programy.	2
2	Typy danych, zmienne, stałe, deklaracje, definicje, operatory arytmetyczne i logiczne, priorytety, rzutowania.	3
3	Instrukcje sterujące (if, switch, pętle while, do-while, for), zakres nazw, czas życia zmiennych.	2
4	Funkcje: definicja, deklaracja, argumenty, wartość zwracana, argumenty domyślne, przeładowanie, rekurencja, wskaźniki do funkcji.	3
5	Tablice, wskaźniki, referencje, dynamiczna alokacja pamięci, struktury i unie.	4
6	Wejście/wyjście, strumienie, operacje na plikach tekstowych i binarnych, preprocesor, przegląd standardowych bibliotek.	4
7	Programowanie obiektowe: tworzenie klas i obiektów, pola i metody, hermetyzacja, konstruktory, destruktory, klasa string.	2
8	Dziedziczenie, hierarchizowanie klas, polimorfizm.	2
9	Szablony funkcji i klas, kontenery STL (vector, list, deque, map, set), iteratory, algorytmy.	3
10	Wyjątki, obsługa błędów, dobre praktyki projektowe, bezpieczeństwo i jakość kodu.	2

Treści programowe		
Lp.	Tematyka zajęć	Liczba godzin
<b>Forma zajęć – laboratorium</b>		
1	Konfiguracja środowiska, kompilacja, pierwsze proste programy w C++.	2
2	Zmienne, operatory, instrukcje sterujące – ćwiczenia w implementacji podstawowej logiki.	2
3	Funkcje i rekurencja – implementacja prostych algorytmów, testowanie poprawności.	2
4	Tablice, wskaźniki, dynamiczna alokacja pamięci – tworzenie prostych struktur danych.	3
5	Struktury, unie, operacje na plikach tekstowych i binarnych.	2
6	Klasy i obiekty – implementacja, enkapsulacja, konstruktory, destruktory.	2
7	Dziedziczenie, szablony, wykorzystanie kontenerów STL w praktyce.	3
8	Obsługa wyjątków, testowanie, debugging, przygotowanie mini-projektu końcowego. Zaliczenie.	2

## AKADEMIA TECHNICZNO-INFORMATYCZNA W NAUKACH STOSOWANYCH

<b>Forma i warunki zaliczenia przedmiotu</b>	Praktyczny egzamin końcowy. Końcowa ocena z laboratoriów zależy od oceny ze sprawdzianu końcowego oraz aktywności w realizacji zagadnień na poszczególnych zajęciach.	
<b>Metody weryfikacji efektów uczenia się</b>		<b>Nr efektu uczenia się z sylabusu</b>
	Egzamin końcowy	W01–W05
	Kolokwium, stopnie częściowe z zadań i aktywności.	U01–U05, K01–K02.

<b>Literatura podstawowa</b>	<ol style="list-style-type: none"> <li>1. J. Grębosz, <i>Opus magnum C++11. Programowanie w języku C++</i>, Helion, Gliwice 2018.</li> <li>2. S. Prata, <i>Język C++. Szkoła programowania</i>. Wydanie VI, Helion, Gliwice 2019.</li> </ol>
<b>Literatura uzupełniająca</b>	<ol style="list-style-type: none"> <li>1. J. Swaminathan, <i>Mastering C++ Programming. Modern C++ 17 at your fingertips</i>, Helion, Gliwice 2017.</li> </ol>

Nakład pracy studenta	
	Liczba godzin
Zajęcia dydaktyczne	45
Przygotowanie się do zajęć	35
Studiowanie literatury	25
Udział w konsultacjach	5
Przygotowanie projektu / eseju / prezentacji itp.	-
Przygotowanie się do egzaminu / zaliczenia	35
Inne	-
<b>ŁĄCZNY nakład pracy studenta w godz.</b>	<b>145</b>
<b>Liczba punktów ECTS</b>	<b>5</b>