

SYLABUS PRZEDMIOTU

1.	Nazwa przedmiotu w języku polskim oraz języku angielskim Zaawansowane programowanie w C++ Advanced C++ Programming
2.	Dyscyplina naukowa informatyka
3.	Język wykładowy polski
4.	Jednostka prowadząca przedmiot Wydział Fizyki i Astronomii
5.	Rodzaj przedmiotu (<i>obowiązkowy, do wyboru</i>) do wyboru
6.	Kierunek studiów Informatyka stosowana i systemy pomiarowe
7.	Poziom studiów I stopień
8.	Rok studiów III rok
9.	Semestr (<i>zimowy lub letni</i>) zimowy
10.	Forma zajęć i liczba godzin wykład – 30 godzin laboratorium komputerowe – 30 godzin
11.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu Umiejętność programowania w C++ na poziomie odpowiadającym kursowi Programowanie w C++, tj. rozumienie co najmniej następujących koncepcji C++: - wyrażenia i instrukcje; - referencje, wskaźniki; - funkcje; - klasy, dziedziczenie, hermetyzacja danych; - elementy biblioteki standardowej C++, w tym iteratory i algorytmy STL; - biblioteki; - preprocessing, kompilacja, łączenie programów.

12.	Cele kształcenia dla przedmiotu Celem zajęć jest opanowanie języka C++ na poziomie zaawansowanym, odpowiadającym typowym wymaganiom na rozmowach kwalifikacyjnych przy zatrudnieniu jako programista C++.	
13.	Treści programowe Zaawansowane koncepcje programowania w C++17: - szablony (w tym metaprogramowanie, szablony wariadyczne); - obsługa wyjątków; - semantyka move; - obiekty funkcyjne i wyrażenia lambda; - programowanie współbieżne w C++17 (thread, mutex, promise/future, async, operacje atomowe; wyścig i zakleszczenie, condition variable). - wybrane standardy programowania współbieżnego w środowisku z pamięcią współdzieloną i rozproszoną, np. OpenMP, AVX, MPI; - inteligentne wskaźniki; - wybrane narzędzia programistyczne związane z C++ (np. cmake, instrumentacja kodu, wybrany analizator, formater, profiler kodu) oraz, w razie potrzeby, obsługa pull requestów w wybranym rozproszonym systemie wersji - testy jednostkowe (np. GTest); - zaawansowane elementy C++17/C++20 (np. structured binding, constexpr, consteval, if constexpr, std::format, std::ranges, std::chrono, liczby pseudolosowe).	
14.	Zakładane efekty uczenia się Student: – zna i rozumie fundamentalne koncepcje języka C++17 powszechnie wykorzystywane w programowaniu przemysłowym, w tym szablony, wyjątki, semantyka move, wyrażenia lambda, metody programowania współbieżnego, inteligentne wskaźniki); – zna na zaawansowanym poziomie wybrane elementy środowiska programistycznego związanego z językiem C++ (np. cmake, instrumentacja kodu, profilowanie kodu); – potrafi napisać program wykorzystując zaawansowane elementy programowania w języku C++.	Symbole odpowiednich kierunkowych efektów uczenia się I1_W04 I1_U06
15.	Literatura obowiązkowa i zalecana Literatura obowiązkowa: Materiały z zajęć dostarczone przez prowadzącego Literatura zalecana: B. Stroustrup, Język C++. Kompendium wiedzy, Helion 2014 C++ Reference, http://en.cppreference.com Serwis StackOverflow, stackoverflow.com Jonathan Boccara, Fluent {C++}, https://www.fluentcpp.com/posts/	

16.	Metody weryfikacji zakładanych efektów uczenia się: ciągła kontrola postępów w zakresie tematyki zajęć poprzez weryfikację rozwiązań zadań z list egzamin (pisemny lub ustny)	
17.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu: ciągła kontrola obecności i postępów w zakresie tematyki zajęć zaliczenie laboratorium komputerowego na podstawie punktów uzyskanych z list zadań egzamin (pisemny lub ustny)	
	Nakład pracy studenta wyrażony w godzinach zajęć oraz punktach ECTS	liczba godzin przeznaczona na zrealizowanie danego rodzaju zajęć
	zajęcia (wg planu studiów) z prowadzącym:	
	– wykład:	30
	– laboratorium:	30
	praca własna studenta (w tym udział w pracach grupowych):	
	– przygotowanie do zajęć:	45
		– czytanie wskazanej literatury:
		15
		– przygotowanie do egzaminu:
		15
Łączna liczba godzin zajęć		135
Liczba punktów ECTS (<i>jeśli jest wymagana</i>)		5