

SYLABUS PRZEDMIOTU

1.	Nazwa przedmiotu w języku polskim oraz angielskim: Programowanie w C++ C++ programming
2.	Dyscyplina: nauki fizyczne
3.	Język wykładowy: polski
4.	Jednostka prowadząca przedmiot: Wydział Fizyki i Astronomii
5.	Rodzaj przedmiotu (obowiązkowy lub do wyboru): obowiązkowy
6.	Kierunek studiów: fizyka, specjalność fizyka komputerowa
7.	Poziom studiów (I stopień lub II stopień): I stopień
8.	Rok studiów: 2
9.	Semestr (zimowy lub letni): letni
10.	Forma zajęć i liczba godzin: wykład - 30, laboratorium komputerowe - 30
11.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu: Student potrafi pisać, kompilować i uruchamiać proste programy w języku C++ lub C.
12.	Cele kształcenia dla przedmiotu: Opanowanie umiejętności tworzenia programów w języku C++ na poziomie średniozaawansowanym, umożliwiającym swobodne korzystanie z bibliotek zewnętrznych (proceduralnych, obiektowych i generycznych). Cele szczegółowe: • Utrwalenie i rozszerzenie kompetencji w zakresie programowania proceduralnego w C++. • Wprowadzenie podstawowych koncepcji programowania obiektowego i generycznego w C++. • Zapoznanie studentów z wybraną biblioteką C++ do tworzenia aplikacji sterowanych zdarzeniami i posiadających nowoczesny graficzny interfejs użytkownika. • Zapoznanie studentów z podstawowymi związkami między oprogramowaniem a sprzętem, na którym wykonywane są programy.
13.	Treści programowe: • Podstawowe koncepcje programowania w C++. ◦ Operatory, wyrażenia i instrukcje. ◦ Funkcje (w tym: argumenty i wartość funkcji, funkcje inline, funkcje składowe klas, polimorfizm nazw funkcji, funkcje rekurencyjne, operator jako funkcja, przeciążanie operatorów, funkcja main). ◦ Typy wbudowane (w tym arytmetyka całkowita i zmiennopozycyjna). ◦ Tablice, wskaźniki i referencje.

	◦ Klasy i obiekty (w tym <code>std::vector</code>, <code>std::string</code>); dziedziczenie; hermetyzacja danych; składowe wirtualne; słowo kluczowe <code>this</code>. ◦ Dynamiczna alokacja pamięci. ◦ Przykładowa dynamiczna struktura danych (np. tablica dynamiczna i/lub lista pojedynczo linkowana). ◦ Strumienie. ◦ Szablony (na poziomie użytkownika). ◦ Wyjątki (na poziomie elementarnym). Wzorzec programistyczny RAII. ◦ Biblioteka STL, w tym iteratory, kontenery i algorytmy (bez szczegółów). ◦ Preprocesor, kompilator, linker. ◦ Używanie bibliotek (w tym pliki nagłówkowe, linkowanie). ◦ Wybrane narzędzia związane z C++, np. środowisko typu QtCreator, debugger; kompilacja programów za pomocą mechanizmu Makefile lub podobnego (np. <code>cmake</code>).	
14.	Zakładane efekty uczenia się: • Zna i rozumie zasady programowania strukturalnego oraz obiektowego w C++. • Zna język C++ w stopniu średniozaawansowanym. • Potrafi posługiwać się wybranym środowiskiem do programowania w C++. • Programuje w języku C++ oraz potrafi tworzyć w nim proste aplikacje użytkowe na systemy klasy PC.	Symbole odpowiednich kierunkowych efektów uczenia się: F1_W09, F1_U07
15.	Literatura obowiązkowa: • Prezentacje z wykładu (udostępniane po kolejnych wykładach) • Notatki z wykładu: https://github.com/zkoza/cpp-issp Literatura zalecana: • Jerzy Grębosz, <i>Opus magnum C++11. Programowanie w języku C++</i>, Helion 2017 • Portal Stack Overflow (stackoverflow.com) • Portal C++ Reference (cppreference.com) • Materiały na YouTube, np. seria kursów C++ „The Cherno”	
16.	Metody weryfikacji zakładanych efektów uczenia się: - egzamin ustny, może być poprzedzony pisemnym testem - oceny rozwiązań list zadań	
17.	Warunki i forma zaliczenia poszczególnych komponentów przedmiotu: - ciągła kontrola obecności i postępów w zakresie tematyki zajęć - egzamin (pisemny lub ustny)	
18.	Nakład pracy studenta	
	Forma realizacji zajęć przez studenta (pozostawić właściwe)	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:	60
	- przygotowanie do sprawdzianów i egzaminu:	15
	Łączna liczba godzin	135
	Liczba punktów ECTS	5

