

OPIS PRZEDMIOTU/MODUŁU KSZTAŁCENIA (SYLABUS)

1.	Nazwa przedmiotu/modułu w języku polskim Programowanie w C++	
2.	Nazwa przedmiotu/modułu w języku angielskim Programming in C++	
3.	Jednostka prowadząca przedmiot Wydział Fizyki i Astronomii	
4.	Kod przedmiotu/modułu 24-ISSP-S1-E2-PC++	
5.	Rodzaj przedmiotu/modułu (<i>obowiązkowy lub fakultatywny</i>) Obowiązkowy	
6.	Kierunek studiów Informatyka stosowana i systemy pomiarowe, Specjalność fizyka komputerowa	
7.	Poziom studiów (<i>I lub II stopień lub jednolite studia magisterskie</i>) I stopień	
8.	Rok studiów (<i>jeśli obowiązuje</i>) 1 ISSP, 2 fizyki komputerowej	
9.	Semestr (<i>zimowy lub letni</i>) letni	
10.	Forma zajęć i liczba godzin Wykład 30 godz., laboratorium 30 godz.	
11.	Imię, nazwisko, tytuł/stopień naukowy osoby prowadzącej zajęcia Zbigniew Koza, dr hab.	
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu oraz zrealizowanych przedmiotów Zaliczony kurs „Praktyczny wstęp do programowania”	
13.	Cele przedmiotu <ul style="list-style-type: none"> • Opanowanie umiejętności tworzenia programów w języku C++ z wykorzystaniem bibliotek zewnętrznych (proceduralnych, obiektowych i generycznych). • Cele szczegółowe: <ul style="list-style-type: none"> ○ Utrwalenie i rozszerzenie kompetencje w zakresie programowania proceduralnego zdobytych na kursie “Praktyczny wstęp do programowania”, ○ Wprowadzenie podstawowych koncepcji programowania obiektowego i generycznego. ○ Przygotowanie bazy dla bardziej zaawansowanych zajęć, w tym “Projekt C++” 	
14.	Zakładane efekty kształcenia Po ukończeniu kursu student: <ul style="list-style-type: none"> • zna: 	Symbole kierunkowych efektów kształcenia, np.:K_W01*, K_U05,K_K03

	<ul style="list-style-type: none"> ○ składnię języka C++ w zakresie potrzebnym do pisania programów proceduralnych i obiektowych oraz używania bibliotek generycznych ○ zasady kompilacji programów w C/C++ (preprocesor, kompilator, linker), w tym programów wykorzystujących biblioteki zewnętrzne ○ podstawowe zasady organizacji pamięci operacyjnej (stos/stereta, inicjalizacja i destrukcja obiektów, zmienne statyczne / automatyczne) ○ sposoby organizacji i przekazywania danych między różnymi fragmentami programu ○ sposoby organizacji kodu w funkcje ○ podstawy programowania obiektowego (dziedziczenie, hermetyzacja danych, funkcje wirtualne) ○ specyfikę arytmetyki stałoprzecinkowej i zmiennoprzecinkowej ○ wybrane elementy biblioteki standardowej C++98 i C++11 ○ podstawowe opcje kompilatora g++ ○ podstawowe sposoby debugowania programów • potrafi: <ul style="list-style-type: none"> ○ pisać i kompilować proste programy w C++, w tym programy korzystające z biblioteki standardowej i bibliotek zewnętrznych ○ pisać i używać w kodzie różne rodzaje podprogramów (funkcje swobodne i metody klas; funkcje <i>inline</i>; metody wirtualne; przeciążone operatory) ○ posługiwać się wybranymi klasami biblioteki standardowej C++ (np. <i>vector</i>, <i>string</i>) ○ korzystać z dokumentacji znajdującej się w Sieci (np. www.cplusplus.com) ○ debugować program` ○ posługiwać się wybranym zintegrowanym środowiskiem programistycznym ○ kompilować program za pomocą mechanizmu Makefile (lub podobnego) 	K_W04, K_W05, K_U03, K_U04, K_U06
15.	<p>Treści programowe</p> <ul style="list-style-type: none"> • Podstawowe koncepcje programowania w C++: <ul style="list-style-type: none"> ○ 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 <i>main</i>) ○ 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>) ○ Dynamiczna alokacja pamięci ○ Dynamiczne struktury danych ○ Strumienie ○ Dziedziczenie i polimorfizm ○ Szablony (na poziomie użytkownika) ○ Wybrane elementy języka C++11 	

	<ul style="list-style-type: none"> ○ Preprocesor, kompilator, linker ● Używanie bibliotek zewnętrznych ● Używanie debugera (wbudowanego w IDE i <i>gdb</i> w trybie tekstowym) ● Kompilacja programów za pomocą mechanizmu Makefile 	
16.	Zalecana literatura (<i>podręczniki</i>) <ol style="list-style-type: none"> 1. Zbigniew Koza, Język C++ Pierwsze starcie, Helion, Gliwice, 2008 2. C++ Reference, http://en.cppreference.com/w/ 3. B. Stroustrup. Język C++, wydanie V, WNT, Warszawa 2000. 4. Jerzy Grębosz, Symfonia C++ Standard, Editions 2000, Kraków 2005 5. Piotr Szawdyński, Kurs C++, http://cpp0x.pl/kursy/Kurs-C++/1 6. http://www.cplusplus.com/ 	
17.	Forma zaliczenia poszczególnych komponentów przedmiotu/modułu, sposób sprawdzenia osiągnięcia zamierzonych efektów kształcenia: wykład: egzamin seminarium: laboratorium: listy zadań; pisemne kolokwium konwersatorium: inne:	
18.	Język wykładowy Polski	
19.	Obciążenie pracą studenta	
	Forma aktywności studenta	Średnia liczba godzin na zrealizowanie aktywności
	Godziny zajęć (wg planu studiów) z nauczycielem:	
	- wykład:	30
	- ćwiczenia:	
	- laboratorium:	30
	- inne:	
	Praca własna studenta np.:	
	- przygotowanie do zajęć:	45
	- opracowanie wyników:	-
	- czytanie wskazanej literatury:	15
	- napisanie raportu z zajęć:	-
	- przygotowanie do egzaminu:	15
	Suma godzin	135
	Liczba punktów ECTS	5

*objaśnienie symboli:

K (przed podkreśleniem) - kierunkowe efekty kształcenia

W - kategoria wiedzy

U - kategoria umiejętności

K (po podkreśleniu) - kategoria kompetencji społecznych

01, 02, 03 i kolejne - numer efektu kształcenia