

WROCŁAWSKA WYŻSZA SZKOŁA INFORMATYKI STOSOWANEJ

KARTA OPISU PRZEDMIOTU

Wydział		Informatyki	
Kierunek		Informatyka	
Specjalność			
Semestr	IV	Program studiów, dla którego obowiązuje sylabus	2019/2020
Stopień studiów	I		

Nazwa przedmiotu	C++ II			
Kod przedmiotu	C++ II			
Łączna liczba godzin	30	Tryb	stacjonarny	niestacjonarny
Profil kształcenia	Ogólnoakademicki (A)		Praktyczny (P)	
Forma zajęć	laboratorium			
Język przedmiotu	polski			
Liczba punktów ECTS	2			

Prowadzący zajęcia	
Forma prowadzonych zajęć	Laboratorium
Wymiar zajęć	30 h
Stopień (tytuł) naukowy	
Imię	
Nazwisko	

Wymagania wstępne	Umiejętność obsługi komputera z systemem Windows. Podstawowa wiedza z zakresu matematyki i logiki. Posiadanie umiejętności algorytmizacji zagadnień. Zaliczenie z przedmiotu C++ I. Posiadanie umiejętności w zakresie czytania i rozumienia programów napisanych w C++. 6. Posiadanie wiedzy w zakresie podstawowych instrukcji. Umiejętność samodzielnego pisania prostych oraz tworzenia nieco bardziej skomplikowanych programów strukturalnych.
Założenia i cele przedmiotu	Kształtowanie umiejętności projektowania, konstruowania i implementacji algorytmów. Kształtowanie umiejętności analizy programów. Poznanie zasad programowania obiektowego i rozwijanie umiejętności czytania kodu programów w C++. Nabycie umiejętności konstruowania dynamicznych struktur danych. Nabycie umiejętności samodzielnego pisania zarówno prostych programów, jak i bardziej złożonych aplikacji obiektowych w języku C++.
Metody dydaktyczne	Zajęcia prowadzone są w formie laboratoriów. Zagadnienia i zadania są objaśnione w scenariuszach zajęć umożliwiającym pracę studenta we własnym tempie. W razie potrzeby dodatkowe przykłady są prezentowane przy tablicy, z wywoływaniem krótkich dyskusji ze studentami, prowokowanych i monitorowanych przez wykładowcę. Główny nacisk położony jest na implementację kolejnych zadań przez studenta przy wsparciu wykładowcy, szczególnie w przypadku pojawienia się problemów. Praca zaliczeniowa/projekt na koniec semestru.

WROCŁAWSKA WYŻSZA SZKOŁA INFORMATYKI STOSOWANEJ

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. Sposoby tworzenia oprogramowania obiektowego, 02. Strukturalny i obiektowy charakter tworzonego oprogramowania. 03. Zaawansowane techniki języka C++, takie jak tworzenie klas i obiektów, posługiwanie się dziedziczeniem i hermetyzacją. 04. Zagrożenia wynikające z nieprawidłowego zastosowania technik omawianych na zajęciach. 05. Bibliotekę STL najpopularniejsze kontenerów. 06. Techniki i narzędzia niezbędne do rozwiązywania prostych problemów inżynierskich z zakresu programowania obiektowego w C++.	K_W01 K_W02 K_W06 K_W07	P6S_WG
UMIEJĘTNOŚCI – absolwent potrafi:	01. Analizować literaturę z zakresu przedmiotu i pozyskiwać informacje potrzebne do realizacji zadań. 02. Rozwiązywać proste zadania z zakresu programowania strukturalnego i obiektowego w C++. 03. Posługiwać się właściwie dobranymi środkami w zakresie implementacji prostych zadań w C++, używając odpowiednich technik i narzędzi. 04. Analizować kod z zakresu podstaw programowania obiektowego w C++. 05. Wyciągać wnioski z analizy własnego kodu. 06. Proponować ulepszenia lub/i usprawnienia zaproponowanego rozwiązania. 07. Prezentować rezultaty otrzymane po uruchomieniu tworzonych przez niego programów. 08. 8. Dyskutuje na temat sposobu realizacji zadań 09. programistycznych	K_U01 K_U02 K_U04 K_U06 K_U08 K_U09 K_U11 K_U15 K_U17	P6S_UW P6S_KK P6S_UO P6S_UK
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

Treści programowe		
Lp.	Tematyka zajęć	Liczba godzin
Forma zajęć – laboratorium		
1	Zajęcia organizacyjne. Powtórzenie i usystematyzowanie wiadomości z poprzedniego semestru.	2
2	Programowanie obiektowe. Tworzenie klas i obiektów. Pola (własności) i metody klas. Klasa string i przetwarzanie napisów.	2
3	Konstruktory i destruktory. Ochrona danych. Kapsułkowanie (hermetyzacja).	3

WROCŁAWSKA WYŻSZA SZKOŁA INFORMATYKI STOSOWANEJ

Treści programowe		
4	Posługiwanie się dziedziczeniem i hierarchizacją.	2
5	Dynamiczne struktury danych. Kontenery sekwencyjne: wektor, lista, deque.	3
6	Set/multiset oraz map jako przykład kontenerów asocjacyjnych. Adaptory stosu i kolejki.	3
7	Iteratory i algorytmy w STL.	2
8	Szablony funkcji i klas.	3
9	Klasy zaprzyjaźnione.	2
10	Wyjątki.	2
11	Zastosowanie poznanych wiadomości do budowy złożonych programów. Programowanie obiektowe w praktyce.	4
12	Repetitorium.	4

Forma i warunki zaliczenia przedmiotu	Praktyczny egzamin końcowy. Końcowa ocena z laboratoriów zależy od oceny ze sprawdzianu końcowego oraz aktywności w realizacji zagadnień na poszczególnych zajęciach.	
Metody weryfikacji efektów uczenia się		Nr efektu uczenia się z sylabusu
	Egzamin końcowy	W01–W06, U01–U09
	Stopnie cząstkowe z zadań i aktywności	W01–W06, U01–U09, K01–K02.

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

Nakład pracy studenta	
	Liczba godzin
Zajęcia dydaktyczne	30
Przygotowanie się do zajęć	10
Studiowanie literatury	5
Udział w konsultacjach	2
Przygotowanie projektu / eseju / prezentacji itp.	-
Przygotowanie się do egzaminu / zaliczenia	10
Inne	-
ŁĄCZNY nakład pracy studenta w godz.	57
Liczba punktów ECTS	2