



karta przedmiotu

1st Podstawowe informacje o przedmiocie

Nazwa przedmiotu	Programowanie
Rocznik studiów	2020/2021
Kolegium	Kolegium Informatyki Stosowanej
Kierunek studiów	Informatyka
Poziom kształcenia	Studia pierwszego stopnia - inżynierskie
Profil kształcenia	Praktyczny
Specjalność	-
Osoba odpowiedzialna	dr Marek Jaszuk

2nd Wymagania wstępne (wynikające z następstwa przedmiotów)

Wstęp do programowania

3. Efekty uczenia się i sposób realizacji zajęć

3.1. Cele przedmiotu

C1	Przekazanie wiedzy w zakresie analizy i opracowania programów komputerowych i implementacji algorytmów
C2	Przekazanie wiedzy w zakresie metodyki i technik programowania oraz rozwiązywania problemów przy pomocy programów komputerowych
C3	Budowanie znajomości i zrozumienia metod, narzędzi, teorii i praktyk stosowanych do projektowania, modelowania i implementacji oprogramowania z uwzględnieniem etapów określenia wymagań, specyfikacji, walidacji i testowania oprogramowania
C4	Kształtowanie umiejętności efektywnego wykorzystywania narzędzi stosowanych w konstruowaniu i dokumentacji procesu wytwarzania oprogramowania, ze szczególnym uwzględnieniem narzędzi do kontroli oprogramowania w tym rozproszonego systemu kontroli wersji
C5	Kształtowanie umiejętności wykorzystania zasobów wielokrotnego użycia w konstruowaniu programów komputerowych
C6	Kształtowanie umiejętności projektowania, implementowania, weryfikowania poprawności i debugowania programów w języku programowania wysokiego poziomu, oraz implementowania algorytmów, wraz z oceną ich złożoności

3.2. Przedmiotowe efekty uczenia się, z podziałem na wiedzę, umiejętności i kompetencje społeczne, wraz z odniesieniem do efektów uczenia się dla kierunku

Lp.	Opis przedmiotowych efektów uczenia się	Odniesienie do efektów uczenia się dla kierunku
Po zaliczeniu przedmiotu student w zakresie wiedzy		
P_W01	ma wiedzę w zakresie matematyki niezbędną do analizy i opracowania	K_W01

	programów komputerowych oraz implementacji algorytmów	
P_W02	ma uporządkowaną wiedzę w zakresie metodyki i technik programowania, w tym technik algorytmicznych oraz znaczenia myślenia algorytmicznego i komputacyjnego w zakresie rozwiązywania problemów przy pomocy programów komputerowych	K_W03
P_W03	zna i rozumie metody, narzędzia, teorie i praktyki stosowane do projektowania, modelowania i implementacji oprogramowania z uwzględnieniem etapów określenia wymagań, specyfikacji, walidacji i testowania oprogramowania	K_W07, K_W14
Po zaliczeniu przedmiotu student w zakresie umiejętności		
P_U01	potrafi efektywnie wykorzystywać narzędzia stosowane w konstruowaniu i dokumentacji procesu wytwarzania oprogramowania, ze szczególnym uwzględnieniem narzędzi do kontroli oprogramowania w tym rozproszonego systemu kontroli wersji	K_U09
P_U02	potrafi świadomie i skutecznie korzystać z zasobów wielokrotnego użycia w konstruowaniu programów komputerowych	K_U10
P_U03	potrafi zaprojektować, zaimplementować, weryfikować poprawność i debugować programy w języku programowania wysokiego poziomu oraz implementować algorytmy, a także ocenić ich złożoność	K_U11

3.3. Formy zajęć dydaktycznych oraz wymiar godzin i punktów ECTS

Studia stacjonarne (ST)							
W	K	Ćw	L	ZP	P	eL	ECTS
20			20				4

Studia niestacjonarne (NST)							
W	K	Ćw	L	ZP	P	eL	ECTS
14			14				4

3.4. Metody realizacji zajęć dydaktycznych

Formy zajęć	Metoda realizacji
Wykład	Wykład informacyjny i problemowy, połączony z elementami demonstracji.
Laboratorium	Ćwiczenia laboratoryjne przy komputerze. W trakcie zajęć studenci dokonują samodzielnej implementacji i debugowania programów komputerowych z wykorzystaniem środowiska zintegrowanego.

3.5. Treści kształcenia (oddzielnie dla każdej formy zajęć)

Wykład

Lp.	Treści kształcenia realizowane w ramach wykładów
W1	Wskaźniki – rola wskaźników w programie
W2	Zarządzanie pamięcią – dynamiczna alokacja pamięci
W3	Wykorzystanie wskaźników do pracy z tablicami i do przesyłania argumentów do funkcji
W4	Referencje – wykorzystanie referencji do przesyłania argumentów do funkcji
W5	Wskaźniki do funkcji
W6	Podstawowe pojęcia z zakresu obiektowości – klasa, obiekt, enkapsulacja
W7	Definiowanie klas i tworzenie obiektów w obiektowym języku programowania, obiekty wartościowe i dynamiczne, struktury
W8	Składniki klasy – pola danych, metody, konstruktory
W9	Kontenery i algorytmy w bibliotece STL
W10	Etapy procesu wytwarzania oprogramowania – specyfikacja wymagań, projektowanie, implementacja, testowanie, wdrożenie, model kaskadowy

Laboratorium

Lp.	Treści kształcenia realizowane w ramach laboratorium
L1	Praca ze wskaźnikami
L2	Dynamiczna alokacja pamięci - tablice dynamiczne jedno- i wielowymiarowe
L3	Przesyłanie argumentów do funkcji – przesyłanie przez wartość, wskaźnik, referencję
L4	Wskaźniki do funkcji
L5	Definiowanie klas i tworzenie obiektów, pliki nagłówkowe
L6	Modyfikatory dostępu, pola danych, metody
L7	Konstruktory, lista inicjalizacyjna, przeładowanie konstruktorów i metod
L8	Obiekty dynamiczne, tablice obiektów
L9	Podstawy pracy z kontenerami i algorytmami w bibliotece STL
L10	Podstawy pracy z systemem kontroli wersji, dokumentowanie kodu przy pomocy komentarzy dokumentujących

3.6. Korelacja pomiędzy efektami uczenia się, celami przedmiotu, a treściami kształcenia

Efekt uczenia się	Cele przedmiotu	Treści kształcenia
P_W01	C1	W1-W9
P_W02	C2	W1-W9
P_W03	C3	W10
P_U01	C4	L10
P_U02	C5	L1-L9
P_U03	C6	L1-L9

3.7. Metody weryfikacji efektów uczenia się

Efekt uczenia się	Metoda oceny	Forma zajęć, w ramach której następuje weryfikacja efektu
P_W01	Egzamin w formie testu	W
P_W02	Egzamin w formie testu	W
P_W03	Egzamin w formie testu	W
P_U01	Kolokwium z laboratorium	L
P_U02	Kolokwium z laboratorium	L
P_U03	Kolokwium z laboratorium	L

3.8. Kryteria oceny stopnia osiągnięcia efektów uczenia się

Efekt uczenia się	Na ocenę 2 student nie potrafi	Na ocenę 3 student potrafi	Na ocenę 4 student potrafi	Na ocenę 5 student potrafi
P_W01	zademonstrować wiedzy w zakresie matematyki niezbędnej do analizy i opracowania programów komputerowych	zademonstrować podstawową wiedzę w zakresie matematyki niezbędną do analizy i opracowania programów komputerowych oraz implementacji algorytmów	zademonstrować na poziomie dobrym wiedzę w zakresie matematyki niezbędną do analizy i opracowania programów komputerowych oraz implementacji algorytmów	zademonstrować na poziomie bardzo dobrym wiedzę w zakresie matematyki niezbędną do analizy i opracowania programów komputerowych oraz implementacji algorytmów
P_W02	zademonstrować wiedzy w zakresie metodyki i technik programowania	zademonstrować wiedzę w zakresie metodyki i technik programowania	zademonstrować wiedzę w zakresie metodyki i technik programowania, w tym technik	zademonstrować wiedzę w zakresie metodyki i technik programowania, w tym

			algorytmicznych	technik algorytmicznych oraz znaczenia myślenia algorytmicznego i komputacyjnego w zakresie rozwiązywania problemów przy pomocy programów komputerowych
P_W03	zademonstrować znajomości i rozumienia metod, narzędzi, teorii i praktyk stosowanych do projektowania, modelowania i implementacji oprogramowania	na poziomie podstawowym zademonstrować znajomość i rozumienie metod, narzędzi, teorii i praktyk stosowanych do projektowania, modelowania i implementacji oprogramowania z uwzględnieniem etapów określenia wymagań, specyfikacji, walidacji i testowania oprogramowania	na poziomie dobrym zademonstrować znajomość i rozumienie metod, narzędzi, teorii i praktyk stosowanych do projektowania, modelowania i implementacji oprogramowania z uwzględnieniem etapów określenia wymagań, specyfikacji, walidacji i testowania oprogramowania	na poziomie bardzo dobrym zademonstrować znajomość i rozumienie metod, narzędzi, teorii i praktyk stosowanych do projektowania, modelowania i implementacji oprogramowania z uwzględnieniem etapów określenia wymagań, specyfikacji, walidacji i testowania oprogramowania
P_U01	efektywnie wykorzystywać narzędzi stosowanych w konstruowaniu i dokumentacji procesu wytwarzania oprogramowania, ze szczególnym uwzględnieniem narzędzi do kontroli oprogramowania w tym rozproszonego systemu kontroli wersji	na poziomie podstawowym wykorzystywać narzędzia stosowane w konstruowaniu i dokumentacji procesu wytwarzania oprogramowania, ze szczególnym uwzględnieniem narzędzi do kontroli oprogramowania w tym rozproszonego systemu kontroli wersji	na poziomie dobrym wykorzystywać narzędzia stosowane w konstruowaniu i dokumentacji procesu wytwarzania oprogramowania, ze szczególnym uwzględnieniem narzędzi do kontroli oprogramowania w tym rozproszonego systemu kontroli wersji	na poziomie bardzo dobrym wykorzystywać narzędzia stosowane w konstruowaniu i dokumentacji procesu wytwarzania oprogramowania, ze szczególnym uwzględnieniem narzędzi do kontroli oprogramowania w tym rozproszonego systemu kontroli wersji
P_U02	korzystać z zasobów wielokrotnego użycia w konstruowaniu programów komputerowych	na poziomie podstawowym korzystać z zasobów wielokrotnego użycia w konstruowaniu programów komputerowych	na poziomie dobrym korzystać z zasobów wielokrotnego użycia w konstruowaniu programów komputerowych	na poziomie bardzo dobrym korzystać z zasobów wielokrotnego użycia w konstruowaniu programów komputerowych
P_U03	zaprojektować, zaimplementować, weryfikować poprawności i debugować programów w języku programowania wysokiego poziomu	zaprojektować, zaimplementować, programy w języku programowania wysokiego poziomu	zaprojektować, zaimplementować, weryfikować poprawności i debugować programy w języku programowania wysokiego poziomu	zaprojektować, zaimplementować, weryfikować poprawności i debugować programy w języku programowania wysokiego poziomu oraz implementować algorytmy, a także ocenić ich złożoność

3.9. Literatura

Literatura podstawowa
Prata S.: Język C++ : szkoła programowania, Helion, Gliwice, 2013 (lub nowsze wydanie)

Stroustrup B.: Język C++. Kompendium wiedzy, Helion, Gliwice 2014 (lub nowsze wydanie)
Stroustrup B.: Programowanie : teoria i praktyka z wykorzystaniem C++, Helion, Gliwice, 2013 (lub nowsze wydanie)
Sacha K.: Inżynieria oprogramowania, Wydawnictwo Naukowe PWN, Warszawa 2010 (lub nowsze wydanie), IBUK Libra

Literatura uzupełniająca
Grębosz J.: Opus magnum C++. Misja w nadprzestrzeń C++14/17, Wydawnictwo Helion, Gliwice 2020
Kubiak M. J., C++ : zadania z programowania z przykładowymi rozwiązaniami, Helion, Gliwice, 2011 lub nowsze

4. Nakład pracy studenta - bilans punktów ECTS

Rodzaje aktywności	Obciążenie studenta	
	studia ST	studia NST
Udział w W/K (UB)	20	14
Konsultacje do W/K (UB)	4	3
Udział w egzaminie z W (UB)	2	2
Samodzielne studiowanie tematyki W/K, w tym przygotowanie do egzaminu/zaliczenia	4	11
Udział w C/L (UB)	20	14
Konsultacje do C/L (UB)	4	3
Samodzielne przygotowanie się do C/L, w tym przygotowanie do zaliczenia	36	43
Sumaryczne obciążenie pracą studenta	90	90
Punkty ECTS za przedmiot	4	4
Punkty ECTS za zajęcia prowadzone z bezpośrednim udziałem nauczycieli i studentów (UB)	2	1
Punkty ECTS za zajęcia kształtujące umiejętności praktyczne (PZ)	2	2