



KARTA PRZEDMIOTU

Kod przedmiotu	studia stacjonarne:	
	studia niestacjonarne:	
Nazwa przedmiotu	Zaawansowane programowanie w języku C++	
Nazwa przedmiotu w języku angielskim	Advanced C++ Programming	
Obowiązuje od roku akademickiego	2022/23	

USYTUOWANIE PRZEDMIOTU W SYSTEMIE STUDIÓW

Kierunek studiów	Informatyka
Poziom kształcenia	I stopień
Profil studiów	ogólnoakademicki
Forma i tryb prowadzenia studiów	studia stacjonarne i niestacjonarne
Zakres	wszystkie specjalności
Jednostka prowadząca przedmiot	Katedra Systemów Informatycznych
Koordynator przedmiotu	dr inż. Grzegorz Łukawski, mgr inż. Paweł Pięta
Zatwierdził	Dziekan Wydziału Elektrotechniki, Automatyki i Informatyki dr hab. inż. Roman Deniziak, prof. PŚk

OGÓLNA CHARAKTERYSTYKA PRZEDMIOTU

Przynależność do grupy/bloku przedmiotów		przedmiot specjalnościowy
Status przedmiotu		wybieralny
Język prowadzenia zajęć		Polski
Usytuowanie w planie studiów - semestr	studia stacjonarne	semestr VI
	studia niestacjonarne	semestr VII
Wymagania wstępne		Programowanie obiektowe 1 Programowanie współbieżne
Egzamin (TAK/NIE)		NIE
Liczba punktów ECTS		4

Forma prowadzenia zajęć		wykład	ćwiczenia	laboratorium	projekt	inne
Liczba godzin w semestrze	studia stacjonarne:	30		30		
	studia niestacjonarne:	18		18		

EFEKTY UCZENIA SIĘ

Kategoria	Symbol efektu	Efekty uczenia się	Odniesienie do efektów kierunkowych
Wiedza	W01	Student zna i rozumie pojęcia i zagadnienia związane z szablonami funkcji i klas oraz standardową biblioteką szablonów (STL) języka C++.	INF1_W18
	W02	Student zna i rozumie zastosowanie wybranych zaawansowanych bibliotek języka C++.	INF1_W18
	W03	Student zna i rozumie pojęcia i zagadnienia związane z tworzeniem zaawansowanych graficznych interfejsów użytkownika w języku C++.	INF1_W18
	W04	Student zna i rozumie zaawansowane pojęcia i zagadnienia związane z programowaniem obiektowym i funkcyjnym w języku C++.	INF1_W18
Umiejętności	U01	Student potrafi tworzyć programy w języku C++ z wykorzystaniem szablonów funkcji i klas oraz standardowej biblioteki szablonów (STL).	INF1_U18
	U02	Student potrafi korzystać z wybranych zaawansowanych bibliotek języka C++.	INF1_U18
	U03	Student potrafi tworzyć zaawansowane graficzne interfejsy użytkownika w języku C++.	INF1_U18
	U04	Student potrafi tworzyć zaawansowane programy w języku C++ w technice obiektowej i funkcyjnej.	INF1_U18
Kompetencje społeczne	K01	Student jest gotów do ciągłego dokształcania się, podnoszenia swoich kompetencji zawodowych i ich krytycznej oceny.	INF1_K01 INF1_K02

TREŚCI PROGRAMOWE

Forma zajęć	Treści programowe
wykład	1. Szablony funkcji i klas. 2. Kontenery i adaptory kontenerów. 3. Iteratory. 4. Algorytmy. 5. Obiekty funkcyjne (funktory). 6. Wybrane zaawansowane biblioteki języka (np. Boost). 7. Zaawansowane operacje wejścia/wyjścia i serializacja obiektów. 8. Programowanie wielowątkowe. 9. Tworzenie zaawansowanych graficznych interfejsów użytkownika. Zaawansowane techniki programowania obiektowego i funkcyjnego.
laboratorium	1. Szablony funkcji i klas. 2. Kontenery i adaptory kontenerów. 3. Iteratory. 4. Algorytmy. 5. Obiekty funkcyjne (funktory). 6. Wybrane zaawansowane biblioteki języka (np. Boost). 7. Zaawansowane operacje wejścia/wyjścia i serializacja obiektów. 8. Programowanie wielowątkowe. 9. Tworzenie zaawansowanych graficznych interfejsów użytkownika. Zaawansowane techniki programowania obiektowego i funkcyjnego.

METODY WERYFIKACJI EFEKTÓW UCZENIA SIĘ

Symbol efektu	Metody sprawdzania efektów uczenia się					
	Egzamin ustny	Egzamin pisemny	Kolokwium	Projekt	Sprawozdanie	Inne
W01			X			
W02			X			
W03			X			
W04			X			
U01			X			X
U02			X			X
U03			X			X
U04			X			X
K01						X

FORMA I WARUNKI ZALICZENIA

Forma zajęć	Forma zaliczenia	Warunki zaliczenia
wykład	zaliczenie z oceną	Uzyskanie co najmniej 50% punktów.
laboratorium	zaliczenie z oceną	Uzyskanie co najmniej 50% punktów.

NAKŁAD PRACY STUDENTA

Bilans punktów ECTS												
Lp.	Rodzaj aktywności	Obciążenie studenta										Jed- nostka
		studia stacjonarne					studia niestacjonarne					
1.	Udział w zajęciach zgodnie z planem studiów	W	C	L	P	S	W	C	L	P	S	h
		30		30			18		18			
2.	Inne (konsultacje, egzamin)	2		2			2		2			h
3.	Razem przy bezpośrednim udziale nauczyciela akademickiego	64					40					h
4.	Liczba punktów ECTS, którą student uzyskuje przy bezpośrednim udziale nauczyciela akademickiego	2,56					1,6					ECTS
5.	Liczba godzin samodzielnej pracy studenta	36					60					h
6.	Liczba punktów ECTS, którą student uzyskuje w ramach samodzielnej pracy	1,44					2,4					ECTS
7.	Nakład pracy związany z zajęciami o charakterze praktycznym	30					18					h
8.	Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym	1,2					0,72					ECTS
9.	Sumaryczne obciążenie pracą studenta	100					100					h
10.	Punkty ECTS za moduł <i>1 punkt ECTS=25 godzin obciążenia studenta</i>	4										ECTS

LITERATURA

1. Scott Meyers, *Effective STL*, 2001.
2. Scott Meyers, *Effective C++*, 3rd Edition, 2005.
3. Scott Meyers, *Effective Modern C++*, 2014.
4. Paul Deitel and Harvey Deitel, *C++ How to Program*, 10th Edition, 2016.
5. David Vandevoorde, Nicolai M. Josuttis and Douglas Gregor, *C++ Templates*, 2nd Edition, 2017.
6. Arthur O'Dwyer, *Mastering the C++17 STL*, 2017.
7. Antony Polukhin, *Boost C++ Application Development Cookbook*, 2nd Edition, 2017.
8. Stroustrup Bjarne, *A Tour of C++*, 2nd Edition, 2018.
9. Ivan Cukic, *Functional Programming in C++*, 2018.
10. Alexandru Bolboaca, *Hands-On Functional Programming with C++*, 2019.
11. Marius Bancila, *Modern C++ Programming Cookbook*, 2nd Edition, 2020.
12. Peter Gottschling, *Discovering Modern C++*, 2nd Edition, 2021.
13. Marc Gregoire, *Professional C++*, 5th Edition, 2021.
14. Paul Deitel and Harvey Deitel, *C++20 for Programmers*, 3rd Edition, 2022.
15. Lee Zhi Eng, *Qt5 C++ GUI Programming Cookbook*, 2nd Edition, 2019.
16. Nibedit Dey, *Cross-Platform Development with Qt 6 and Modern C++*, 2021.
17. Standard C++ (<https://isocpp.org>).
18. The C++ Reference (<https://en.cppreference.com>).
19. The C++ Resources Network (<http://www.cplusplus.com>).
20. Biblioteka Boost (<https://www.boost.org>).
21. Materiały dostępne na platformie NetAcad (<https://www.netacad.com>).