



IV. Opis programu studiów

3. KARTA PRZEDMIOTU

Kod przedmiotu	Brak przedmiotu w systemie USOS
Nazwa przedmiotu	Programowanie w języku C++ 2
Nazwa przedmiotu w języku angielskim	Programming in C++ 2
Obowiązuje od roku akademickiego	2019/20

USYTUOWANIE MODUŁU W SYSTEMIE STUDIÓW

Kierunek studiów	
Poziom kształcenia	
Profil studiów	
Forma i tryb prowadzenia studiów	
Zakres	
Jednostka prowadząca przedmiot	
Koordynator przedmiotu	dr hab. inż. Roman Stanisław Deniziak, prof. PŚk mgr inż. Paweł Pięta
Zatwierdził	Dziekan Wydziału Elektrotechniki Automatyki i Informatyki Dr hab. inż. Antoni Różowicz, prof. PŚk

OGÓLNA CHARAKTERYSTYKA PRZEDMIOTU

Przynależność do grupy/bloku przedmiotów	
Status przedmiotu	
Język prowadzenia zajęć	
Usytuowanie modułu w planie studiów - semestr	
Wymagania wstępne	Programowanie w języku C++ 1
Egzamin (TAK/NIE)	
Liczba punktów ECTS	

Forma prowadzenia zajęć	wykład	ćwiczenia	laboratorium	projekt	inne
Liczba godzin w semestrze	15	0	30	30	0

EFEKTY UCZENIA SIĘ

Kategoria	Symbol efektu	Efekty kształcenia	Odniesienie do efektów kierunkowych
Wiedza	W01	Wiedza na temat zagadnień związanych z wykorzystaniem szablonów funkcji i klas w języku C++.	INF1_W11
	W02	Wiedza na temat zagadnień związanych z wykorzystaniem standardowej biblioteki szablonów (STL) języka C++.	INF1_W11
	W03	Wiedza na temat sposobów rozwiązywania problemów programistycznych z zastosowaniem szablonów funkcji i klas oraz standardowej biblioteki szablonów (STL) języka C++.	INF1_W11
	W04	Wiedza na temat zagadnień związanych z zaawansowanymi operacjami wejścia/wyjścia w języku C++ (STL I/O).	INF1_W11
Umiejętności	U01	Umiejętność tworzenia programów w języku C++ z wykorzystaniem szablonów funkcji i klas.	INF1_U19
	U02	Umiejętność tworzenia programów w języku C++ z wykorzystaniem standardowej biblioteki szablonów (STL).	INF1_U19
	U03	Umiejętność rozwiązywania problemów programistycznych z zastosowaniem szablonów funkcji i klas oraz standardowej biblioteki szablonów (STL) języka C++.	INF1_U19
	U04	Umiejętność tworzenia programów w języku C++ z wykorzystaniem zaawansowanych operacji wejścia/wyjścia (STL I/O).	INF1_U19
	U05	Umiejętność pracy w zespole podczas rozwiązywania problemów programistycznych i umiejętność ich podziału na mniejsze zadania oraz opracowania harmonogramu pracy zapewniającego dotrzymanie terminów.	INF1_U02 INF1_U32
	U06	Umiejętność opracowania dokumentacji i sprawozdania dotyczących rozwiązane go problemu programistycznego oraz przygotowania prezentacji poświęconej wynikom realizacji zadania.	INF1_U03 INF1_U04
Kompetencje społeczne	K01	Świadomość potrzeby i możliwości ciągłego dokształcania się, podnoszenia swoich kompetencji zawodowych.	INF1_K01
	K02	Świadomość odpowiedzialności za pracę własną oraz gotowość podporządkowania się zasadom pracy w zespole i ponoszenia odpowiedzialności za wspólnie realizowane zadania.	INF1_K03

TREŚCI PROGRAMOWE

Forma zajęć*	Treści programowe
wykład	1. Wstęp do standardowej biblioteki szablonów (STL).
	2. Kontener vector.
	3. Kontenery deque i list.
	4. Iteratory.
	5. Podstawowe metody kontenerów.
	6. Adaptery kontenerów.
	7. Kontenery set i multiset.
	8. Kontenery map i multimap.
	9. Algorytmy w STL: operacje niemodyfikujące.
	10. Algorytmy w STL: operacje modyfikujące.
	11. Algorytmy w STL: operacje sortujące.
	12. Algorytmy w STL: operacje na posortowanych ciągach i operacje min/max.
	13. Obiekty funkcyjne.

	14. Zaawansowane operacje wejścia/wyjścia (STL I/O).
	15. Szablony funkcji i klas.
laboratorium	1. Wstęp do standardowej biblioteki szablonów (STL).
	2. Kontener vector.
	3. Kontenery deque i list.
	4. Iteratory.
	5. Podstawowe metody kontenerów.
	6. Adaptery kontenerów.
	7. Kontenery set i multiset.
	8. Kontenery map i multimap.
	9. Algorytmy w STL: operacje niemodyfikujące.
	10. Algorytmy w STL: operacje modyfikujące.
	11. Algorytmy w STL: operacje sortujące.
	12. Algorytmy w STL: operacje na posortowanych ciągach i operacje min/max.
	13. Obiekty funkcyjne.
	14. Zaawansowane operacje wejścia/wyjścia (STL I/O).
	15. Szablony funkcji i klas.
projekt	Zadania projektowe wykonywane w zespołach, polegające na zaimplementowaniu aplikacji w języku C++ z zastosowaniem paradygmatu programowania obiektowego. Powstały program powinien wykorzystywać szablony funkcji i klas oraz standardową bibliotekę szablonów (STL). Aplikacja powinna posiadać intuicyjny interfejs graficzny.

*) zostawić tylko realizowane formy zajęć

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	X	X
U02			X	X	X	X
U03			X	X	X	X
U04			X	X	X	X
U05				X	X	X
U06				X	X	X
K01				X	X	X
K02				X	X	X

A.

FORMA I WARUNKI ZALICZENIA

Forma zajęć*	Forma zaliczenia	Warunki zaliczenia
wykład		Uzyskanie co najmniej 70% punktów.
laboratorium		Uzyskanie co najmniej 70% punktów.
projekt		Dostateczne wykonanie projektu (wraz ze sprawozdaniem i dokumentacją kodu źródłowego) i jego pozytywna obrona.

*) zostawić tylko realizowane formy zajęć

NAKŁAD PRACY STUDENTA

Bilans punktów ECTS							
Lp.	Rodzaj aktywności	Obciążenie studenta					Jednostka
		W	C	L	P	S	
1.	Udział w zajęciach zgodnie z planem studiów	15		30	30		h
3.	Inne (konsultacje, egzamin)*	2		2	2		h
4.	Razem przy bezpośrednim udziale nauczyciela akademickiego	81					h
5.	Liczba punktów ECTS, którą student uzyskuje przy bezpośrednim udziale nauczyciela akademickiego	3,24					ECTS
6.	Liczba godzin samodzielnej pracy studenta	69					h
7.	Liczba punktów ECTS, którą student uzyskuje w ramach samodzielnej pracy	2,76					ECTS
8.	Nakład pracy związany z zajęciami o charakterze praktycznym	60					h
9.	Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym	2,4					ECTS
10.	Sumaryczne godzinowe obciążenie pracą studenta	150					h
11.	Punkty ECTS za moduł <i>1 punkt ECTS=25 godzin obciążenia studenta</i>	6					

* wszelkie formy weryfikacji efektów, w tym egzaminy oraz nie więcej niż 2 godziny konsultacji dla każdej formy zajęć

LITERATURA

1. Materiały dostępne na platformie NetAcad (www.netacad.com).

·
·
·
·
·

Uwaga: wykaz literatury winien uwzględniać aktualne i dostępne publikacje