



Karta przedmiotu

Nazwa i kod przedmiotu	Programming, PG_00046329						
Kierunek studiów	Elektronika i telekomunikacja (studia w jęz. angielskim), Informatyka (studia w jęz. angielskim), Automatyka, cybernetyka i robotyka (studia w jęz. angielskim)						
Data rozpoczęcia studiów	październik 2023 r.	Rok akademicki realizacji przedmiotu			2023/2024		
Poziom kształcenia	II stopnia	Grupa zajęć			Grupa zajęć obowiązkowych z zakresu kierunku studiów		
Forma studiów	stacjonarne	Sposób realizacji			na uczelni		
Rok studiów	1	Język wykładowy			angielski		
Semestr studiów	1	Liczba punktów ECTS			5.0		
Profil kształcenia	ogólnoakademicki	Forma zaliczenia			egzamin		
Jednostka prowadząca	Wydział Elektroniki, Telekomunikacji i Informatyki -> Katedra Algorytmów i Modelowania Systemów						
Imię i nazwisko wykładowcy (wykładowców)	Odpowiedzialny za przedmiot		dr Marcin Jurkiewicz				
	Prowadzący zajęcia z przedmiotu		dr Marcin Jurkiewicz				
Formy zajęć i metody nauczania	Forma zajęć	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium	RAZEM
	Liczba godzin zajęć	15.0	0.0	15.0	10.0	0.0	40
	W tym liczba godzin zajęć na odległość: 0.0						
Aktywność studenta i liczba godzin pracy	Aktywność studenta	Udział w zajęciach dydaktycznych, objętych planem studiów		Udział w konsultacjach		Praca własna studenta	RAZEM
	Liczba godzin pracy studenta	40		20.0		65.0	125
Cel przedmiotu	Celem przedmiotu jest nabycie przez studentów wiedzy oraz umiejętności w zakresie programowania i implementacji programów w środowisku Linux/Visual Studio. Student powinien opanować podstawowy repertuar dostępnych w języku C/C++ instrukcji, typów danych, operatorów oraz funkcji. Student powinien nabyć wiedzę na temat struktur, wskaźników oraz innych podstawowych pojęć związanych z programowaniem w języku C/C++.						
Efekty uczenia się przedmiotu	Efekt kierunkowy		Efekt z przedmiotu		Sposób weryfikacji i oceny efektu		
	[K7_W04] zna i rozumie w zaawansowanym stopniu zasady, metody i techniki programowania oraz zasady tworzenia oprogramowania komputerów albo programowania urządzeń lub sterowników wykorzystujących mikroprocesory albo inne elementy lub układy programowalne, specyficznych dla kierunku studiów, a także organizację pracy systemów wykorzystujących komputery lub te urządzenia		Student zna podstawowe zasady programowania języka C/C++.		[SW1] Ocena wiedzy faktograficznej		
	[K7_U04] potrafi wykorzystywać posiadaną wiedzę z zakresu metod i technik programowania oraz dobrać i zastosować właściwe metody i narzędzia programistyczne w tworzeniu oprogramowania komputerów albo programowania urządzeń lub sterowników wykorzystujących mikroprocesory albo elementy lub układy programowalne, charakterystycznych dla danego kierunku studiów, dokonując oceny i krytycznej analizy wykonanego oprogramowania, a także syntezy i twórczej interpretacji prezentowanych za jego pomocą informacji		Student potrafi wykorzystać przekazaną wiedzę (z wykładu), podstawowe techniki języka C/C++ oraz oprogramowanie w systemie Linux/Visual Studio do napisania i kompilacji programu.		[SU1] Ocena realizacji zadania [SU4] Ocena umiejętności korzystania z metod i narzędzi		

Treści przedmiotu	<ol style="list-style-type: none"> <li>1. Języki programowania, alfabet, syntaktyka i semantyka. Translacja.</li> <li>2. Klasyfikacja typów. Typy całkowite i zmiennoprzecinkowe.</li> <li>3. Operatory i wyrażenia arytmetyczne.</li> <li>4. Wybrane funkcje standardowe.</li> <li>5. Typ znakowy. Rzutowanie typów.</li> <li>6. Typ logiczny. Operatory i wyrażenia logiczne.</li> <li>7. Podstawy obsługi wejścia/wyjścia.</li> <li>8. Instrukcje warunkowe (if, switch) oraz wyrażenie warunkowe.</li> <li>9. Instrukcje iteracyjne (for, while, do-while). Iteracje zagnieżdżone.</li> <li>10. Definiowanie typów. Stałe. Typ wyliczeniowy.</li> <li>11. Tablice jednowymiarowe i wielowymiarowe. Napisy.</li> <li>12. Zakres ważności i czas życia zmiennych</li> <li>13. Funkcje. Zasięg i czas życia zmiennych. Efekt uboczny.</li> <li>14. Przesyłanie parametrów funkcji.</li> <li>15. Typ wskaźnikowy. Arytmetyka wskaźników.</li> <li>16. Wskaźniki w komunikacji między funkcjami.</li> <li>17. Dynamiczna alokacja pamięci.</li> <li>18. Struktury</li> </ol>		
Wymagania wstępne i dodatkowe	Brak wymagań		
Sposoby i kryteria oceniania osiągniętych efektów uczenia się	Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej
	Laboratorium: poprawność działania programu, algorytmy, struktury danych, czas wykonania oraz uniwersalność programu.	50.0%	30.0%
	Projekt: poprawność działania programu, algorytmy, struktury danych, czas wykonania oraz uniwersalność programu.	50.0%	30.0%
	Egzamin	50.0%	40.0%
Zalecana lista lektur	Podstawowa lista lektur	1. KERNIGHAN, Brian W.; RITCHIE, Dennis M. <i>The C programming language</i> , Prentice Hall, 2006	
	Uzupełniająca lista lektur	1. B. Stroustrup, <i>The C++ Programming Language</i> , Addison Wesley Longman, 2000	
	Adresy eZasobów	Adresy na platformie eNauczanie: 2023 Programming ABC - Moodle ID: 33816 <a href="https://enauczanie.pg.edu.pl/moodle/course/view.php?id=33816">https://enauczanie.pg.edu.pl/moodle/course/view.php?id=33816</a>	
Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania	<p><b>Co pokaże poniższy fragment kodu, jeśli będzie częścią poprawnego programu?</b></p> <pre>int i;  for(i=0;i&lt;3;i++);  cout&lt;  cout&lt;</pre> <p>a) 011223 b) 0124 c) 0123 d) 34 e) 124 f) 45</p>		
Praktyki zawodowe w ramach przedmiotu	Nie dotyczy		