



Karta przedmiotu

Nazwa i kod przedmiotu	Współczesne środowiska programowania, PG_00053918						
Kierunek studiów	Automatyka, cybernetyka i robotyka						
Data rozpoczęcia studiów	październik 2022 r.	Rok akademicki realizacji przedmiotu			2024/2025		
Poziom kształcenia	I stopnia - inżynierskie	Grupa zajęć			Grupa zajęć fakultatywnych Grupa zajęć powiązanych z prowadzonymi badaniami naukowymi w dziedzinie nauki związanej z kierunkiem - profil ogólnoakademicki		
Forma studiów	stacjonarne	Sposób realizacji			na uczelni		
Rok studiów	3	Język wykładowy			polski		
Semestr studiów	5	Liczba punktów ECTS			3.0		
Profil kształcenia	ogólnoakademicki	Forma zaliczenia			zaliczenie		
Jednostka prowadząca	Wydział Elektroniki -> Telekomunikacji i Informatyki -> Katedra Systemów Decyzyjnych i Robotyki						
Imię i nazwisko wykładowcy (wykładowców)	Odpowiedzialny za przedmiot		dr hab. inż. Tomasz Stefański				
	Prowadzący zajęcia z przedmiotu						
Formy zajęć i metody nauczania	Forma zajęć	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium	RAZEM
	Liczba godzin zajęć	15.0	0.0	15.0	0.0	0.0	30
	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	30		4.0		41.0	75
Cel przedmiotu	Zapoznanie studentów ze współczesnymi środowiskami programowania na przykładzie technologii CUDA firmy Nvidia.						

Efekty uczenia się przedmiotu	Efekt kierunkowy	Efekt z przedmiotu	Sposób weryfikacji i oceny efektu
	[K6_W03] zna i rozumie w zaawansowanym stopniu budowę i zasady działania komponentów i systemów związanych z kierunkiem studiów, w tym teorie, metody i złożone zależności między nimi oraz wybrane zagadnienia szczegółowe – właściwe dla programu kształcenia	Student potrafi pisać programy w językach C/C++ stosujące karty graficzne.	[SW1] Ocena wiedzy faktograficznej
	[K6_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 elementy lub układy programowalne, specyficznych dla kierunku studiów, a także organizację pracy systemów wykorzystujących komputery lub te urządzenia	Student rozumie zasady programowania równoległego i potrafi pisać programy stosujące karty graficzne.	[SW1] Ocena wiedzy faktograficznej
	[K6_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	Student potrafi pisać programy inżynierskie w środowisku CUDA.	[SU1] Ocena realizacji zadania [SU4] Ocena umiejętności korzystania z metod i narzędzi
	[K6_W01] zna i rozumie w zaawansowanym stopniu matematykę w zakresie niezbędnym do formułowania i rozwiązywania prostych zagadnień związanych z kierunkiem studiów	Student potrafi pisać programy przetwarzające równoległe duże ilości danych.	[SW1] Ocena wiedzy faktograficznej
Treści przedmiotu	<ol style="list-style-type: none"> 1. Wprowadzenie do architektury i środowiska programowania CUDA. 2. Omówienie kompilatora (nvcc), debugera (cuda-gdb) oraz profilera w środowisku CUDA. 3. Model programowania równoległego: jądra obliczeniowe, wątki, bloki wątków, hierarchia pamięci, programowanie heterogeniczne, możliwości obliczeniowe kart graficznych firmy NVIDIA. 4. Interfejs programowania CUDA C. 5. Optymalizacja kodu. 6. Przegląd bibliotek programistycznych dostępnych w ramach środowiska CUDA. 7. Wprowadzenie do standardu OpenCL. 		
Wymagania wstępne i dodatkowe	Student posiada podstawową wiedzę z programowania.		
Sposoby i kryteria oceniania osiągniętych efektów uczenia się	Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej
	wykład	40.0%	20.0%
	laboratorium	40.0%	30.0%
	projekt	40.0%	50.0%
Zalecana lista lektur	Podstawowa lista lektur	<ul style="list-style-type: none"> - CUDA C Programming Guide - CUDA Runtime API - CUDA C Best Practices Guide 	
	Uzupełniająca lista lektur	<ul style="list-style-type: none"> - CUDA Compiler Driver NVCC - CUDA Visual Profiler - CUDA-gdb debugger - CUDA-memcheck memory checker 	
	Adresy eZasobów		
Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania	<ol style="list-style-type: none"> 1. Model programowania równoległego CUDA. 2. Rodzaje pamięci na karcie graficznej. 3. Zastosowanie pamięci współdzielonej. 4. Strumienie w środowisku CUDA. 5. Zdarzenia w środowisku CUDA. 		
Praktyki zawodowe w ramach przedmiotu	Nie dotyczy		