

KARTA KURSU

Nazwa	Programowanie na GPU
Nazwa w j. ang.	GPU Programming

Koordinator	mgr Michał Frontczak	Zespół dydaktyczny
		mgr Michał Frontczak
Punktacja ECTS*	st. stacjonarne: 2 st. niestacjonarne: 2	

Opis kursu (cele kształcenia)

Celem kursu jest nauczenie studentów umiejętności programowania z wykorzystaniem technologii NVIDIA CUDA.

Kurs prowadzony jest w języku polskim.

Podstawowa znajomość języka angielskiego jest wymagana w celu rozumienia dokumentacji technicznej oraz materiałów dydaktycznych.

Warunki wstępne

Wiedza	Wiedza na poziomie studiów licencjackich lub inżynierskich.
Umiejętności	Potrafi zapisywać podstawowe algorytmy i definiować struktury danych za pomocą języka C++ i Python.
Kursy	Wstępne kursy nie są wymagane.

Efekty uczenia się

	Efekt uczenia się dla kursu	Odniesienie do efektów kierunkowych
Wiedza	Po zakończeniu kursu student:	
	W01: Posiada wiedzę z programowania i testowania z wykorzystaniem GPU.	K_W03
	W02: Posiada wiedzę na temat struktur danych oraz algorytmów wykorzystywanych w złożonych obliczeniach wykonywanych na GPU.	K_W04
	W03: Posiada wiedzę na temat rozwiązywania problemów optymalizacji obliczeń z wykorzystaniem GPU.	K_W08

	Efekt uczenia się dla kursu	Odniesienie do efektów kierunkowych
Umiejętności	Po zakończeniu kursu student:	
	U01: umie uzasadniać wybór konkretnych rozwiązań programistycznych w odniesieniu do realizowanych zadań uwzględniając ich złożoność, wydajność oraz możliwości rozwoju aplikacji.	K_U02, K_U04
	U02: potrafi wykorzystać programowanie na GPU w celu optymalizacji modeli matematycznych.	K_U06

	U03: umie znajdować i wykorzystywać informacje potrzebne do rozszerzania swojej wiedzy z zakresu programowania z wykorzystaniem kart graficznych.	K_U09
	U04: potrafi omówić oraz zaprezentować zastosowane algorytmy, ich złożoność, udokumentować, uzasadnić zastosowane algorytmy w projekcie.	K_U10

Kompetencje społeczne	Efekt uczenia się dla kursu	Odniesienie do efektów kierunkowych
	Po zakończeniu kursu student:	
	K01: potrafi korzystać z różnych źródeł informacji (w tym zasobów sieciowych oraz dokumentacji technicznej) do poszerzania własnej wiedzy i zdobywania nowych umiejętności	K_K01

Studia stacjonarne

Organizacja											
Forma zajęć	Wykład (W)	Ćwiczenia w grupach									
		A		K		L		S		P	
Liczba godzin						20					

Studia niestacjonarne

Organizacja												
Forma zajęć	Wykład (W)	Ćwiczenia w grupach										
		A		K		L		S		P		E
Liczba godzin					15							

Opis metod prowadzenia zajęć

Kurs prowadzony jest w formie ćwiczeń laboratoryjnych. Studenci samodzielnie implementują zadania programistyczne/algorytmiczne uruchamiane na GPU. Ponadto studenci otrzymują poprzez platformę e-learningową zestawy problemów do samodzielnego rozwiązania.

Formy sprawdzania efektów uczenia się

	E – learning	Gry dydaktyczne	Ćwiczenia w szkole	Zajęcia terenowe	Praca laboratoryjna	Projekt indywidualny	Projekt grupowy	Udział w dyskusji	Referat	Praca pisemna (esej)	Egzamin ustny	Egzamin pisemny	Inne
W01	X				X	X		X					
W02	X				X	X		X					
W03	X				X	X		X					
U01					X	X							
U02					X	X							
U03					X	X							
U04					X	X							
K01					X								

Kryteria oceny	<p>Warunkiem otrzymania zaliczenia laboratorium jest realizacja wszystkich zadań laboratoryjnych oraz przedstawienie certyfikatu ukończenia szkolenia na platformie e-learningowej.</p> <p>Ocena będzie wystawiana na podstawie projektu końcowego:</p> <ul style="list-style-type: none"> • Prezentacja modelu fizycznego lub matematycznego. • Sposób jego optymalizacji z wykorzystaniem GPU. • Dokumentacja.
----------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Uwagi	
-------	--

Treści merytoryczne (wykaz tematów)

<ol style="list-style-type: none"> 1. Architektura GPU 2. Wstęp do programowania CUDA 3. Zarządzanie pamięcią i model programistyczny 4. Wątki, strumień w CUDA 5. Debugowanie i profilowanie 6. Integracja Pythona i CUDA (biblioteka Numba)

Wykaz literatury podstawowej

<ol style="list-style-type: none"> 1. CUDA C++ Programming Guide, https://docs.nvidia.com/cuda/cuda-c-programming-guide/ 2. Programming Massively Parallel Processors, David B. Kirk, Wen-mei W. Hwu, Morgan Kaufmann 2016

Wykaz literatury uzupełniającej

<ol style="list-style-type: none"> 1. Marek Sawerwain, OpenCL. Akceleracja GPU w praktyce, PWN 2014 2. Jason Sanders, Edward Kandrot, CUDA w przykładach. Wprowadzenie do ogólnego programowania procesorów GPU, Helion 2012

Bilans godzinowy zgodny z CNPS (Całkowity Nakład Pracy Studenta) - **studia stacjonarne**

liczba godzin w kontakcie z prowadzącymi	Wykład	0
	Konwersatorium (ćwiczenia, laboratorium itd.)	20
	Pozostałe godziny kontaktu studenta z prowadzącym	10
liczba godzin pracy studenta bez kontaktu z prowadzącymi	Lektura w ramach przygotowania do zajęć	20
	Przygotowanie krótkiej pracy pisemnej lub referatu po zapoznaniu się z niezbędną literaturą przedmiotu	0
	Przygotowanie projektu lub prezentacji na podany temat (praca w grupie)	10
	Przygotowanie do egzaminu/zaliczenia	0
Ogółem bilans czasu pracy		60
Liczba punktów ECTS w zależności od przyjętego przelicznika		2

Bilans godzinowy zgodny z CNPS (Całkowity Nakład Pracy Studenta) - **studia niestacjonarne**

liczba godzin w kontakcie z prowadzącymi	Wykład	0
	Konwersatorium (ćwiczenia, laboratorium itd.)	15
	Pozostałe godziny kontaktu studenta z prowadzącym	5
liczba godzin pracy studenta bez kontaktu z prowadzącymi	Lektura w ramach przygotowania do zajęć	25
	Przygotowanie krótkiej pracy pisemnej lub referatu po zapoznaniu się z niezbędną literaturą przedmiotu	0
	Przygotowanie projektu lub prezentacji na podany temat (praca w grupie)	10
	Przygotowanie do egzaminu/zaliczenia	0
Ogółem bilans czasu pracy		60
Liczba punktów ECTS w zależności od przyjętego przelicznika		2