

1.	Nazwa kierunku	informatyka
2.	Cykl rozpoczęcia	2014/2015 (semestr letni)
3.	Poziom kształcenia	studia drugiego stopnia
4.	Profil kształcenia	ogólnoakademicki
5.	Forma prowadzenia studiów	stacjonarna

Moduł kształcenia: Programowanie współbieżne

Kod modułu: 08-IN-IJO-S2-PW

1. Liczba punktów ECTS: 4

2. Zakładane efekty kształcenia modułu			
kod	opis	efekty kształcenia kierunku	stopień realizacji (skala 1-5)
PW -K_8	Potrafi pracować w zespole dwuosobowym i dokonuje właściwego podziału pracy	K_2_A_I_K01	1
		K_2_A_I_K03	1
		K_2_A_I_K06	1
PW -U_4	Potrafi dokonać dekompozycji rozpatrywanego problemu obliczeniowego na składowe umożliwiające opracowanie algorytmu równoległego.	K_2_A_I_U13	2
PW -U_5	Potrafi zastosować mechanizmy takie jak zmienne i instrukcje atomowe, semaforey, monitory w implementacji programów współbieżnych.	K_2_A_I_U13	2
PW -U_6	Potrafi ocenić efektywność algorytmów równoległych za pomocą podstawowych miar, takich jak przyspieszenie, koszt i efektywność. Potrafi ocenić możliwe do uzyskania przyspieszenie na podstawie prawa Amdahla oraz Gustavsona.	K_2_A_I_U13	2
PW -U_7	Potrafi rozwiązać problem wzajemnego wykluczania w środowisku rozproszonym, a także zaimplementować algorytm uzgadniania rozproszonego.	K_2_A_I_U13	2
PW -W_1	Ma wiedzę na temat architektury programowalnych układów graficznych (GPU) i narzędzi programistycznych umożliwiającą tworzenie dedykowanego oprogramowania.	K_2_A_I_W04	2
		K_2_A_I_W07	2
		K_2_A_I_W09	1
PW -W_2	Ma wiedzę na temat wzorców komunikacji i typowych operacji zbiorczych w programach równoległych ze szczególnym uwzględnieniem GPU. Zna algorytmy równoległe takie jak redukcja, suma prefiksowa, histogram, wybrane algorytmy sortowania.	K_2_A_I_W06	3
PW -W_3	Ma wiedzę na temat modeli komputerów z pamięcią rozproszoną. Zna zasady programowania komputerów z pamięcią rozproszoną.	K_2_A_I_W06	3

3. Opis modułu	
Opis	Celem zajęć jest przygotowanie studentów do projektowania i implementacji poprawnych i efektywnych algorytmów współbieżnych ze szczególnym uwzględnieniem programowalnych układów graficznych oraz komputerów równoległych z pamięcią rozproszoną. W trakcie zajęć studenci uzyskają wiedzę na temat typowych problemów związanych z projektowaniem poprawnych i efektywnych programów równoległych, a także wybranych sposobów ich rozwiązywania. Dodatkowo studenci zdobędą niezbędną wiedzę dotyczącą podstaw projektowania efektywnych algorytmów równoległych.
Wymagania wstępne	Dobra znajomość programowania w języku C i elementarna znajomość C++. Znajomość obsługi zintegrowanych środowisk programistycznych i podstawowych narzędzi programistycznych wspomagających tworzenie i testowanie oprogramowania.

4. Sposoby weryfikacji efektów kształcenia modułu			
kod	nazwa (typ)	opis	efekty kształcenia modułu
PW_w_1	Prace kontrolne	Co najmniej jedno kolokwium sprawdzające wiedzę z tematyki poruszanej na wykładzie oraz realizowanej w ramach laboratoriów.	PW -U_4, PW -U_5, PW -U_6, PW -U_7, PW -W_1, PW -W_2, PW -W_3
PW_w_2	Projekt programistyczny	Realizacja projektu / projektów programistycznych pozwalających na praktyczną weryfikację nabytej wiedzy i umiejętności dotyczących programowania współbieżnego, ze szczególnym uwzględnieniem programowania układów graficznych oraz programowania komputerów z pamięcią rozproszoną.	PW -K_8, PW -U_5, PW -U_6, PW -U_7
PW_w_3	Egzamin	Test złożony z pytań wielokrotnego wyboru oraz zadań otwartych dotyczących tematów poruszanych na wykładach oraz laboratoriach.	PW -U_4, PW -U_5, PW -U_6, PW -U_7, PW -W_1, PW -W_2, PW -W_3

5. Rodzaje prowadzonych zajęć						
kod	rodzaj prowadzonych zajęć			praca własna studenta		sposoby weryfikacji efektów kształcenia
	nazwa	opis (z uwzględnieniem metod dydaktycznych)	liczba godzin	opis	liczba godzin	
PW_fs_1	wykład	Podanie treści kształcenia w formie werbalnej z wykorzystaniem wizualizacji treści. Omówienie ważnych kwestii teoretycznych i praktycznych dotyczących programowania współbieżnego, ze szczególnym uwzględnieniem programowania równoległego.	30	Zapoznanie się z tematyką prezentowaną podczas wykładów, w tym lektura literatury obowiązkowej oraz uzupełniającej.	15	PW_w_1
PW_fs_2	laboratorium	Szczegółowe przygotowanie studentów do tworzenia aplikacji współbieżnych w nowoczesnych językach programowania. Prezentacja i omówienie narzędzi wspierających realizację oprogramowania współbieżnego.	30	Rozwiązywanie zadań praktycznych z poszczególnych tematów wraz z analizą rozwiązań już istniejących. Zrealizowanie projektu programistycznego z zastosowaniem prezentowanych na wykładach metod.	45	PW_w_2, PW_w_3