

KARTA OPISU MODUŁU KSZTAŁCENIA		
Nazwa modułu/przedmiotu Programowanie wieloparadygmatowe		Kod 1010334591010337136
Kierunek studiów Informatyka	Profil kształcenia (ogólnoakademicki, praktyczny) (brak)	Rok / Semestr 5 / 9
Ścieżka obieralności/specjalność Technologie informatyczne	Przedmiot oferowany w języku: polski	Kurs (obligatoryjny/obieralny) obligatoryjny
Stopień studiów: I stopień	Forma studiów (stacjonarna/niestacjonarna) niestacjonarna	
Godziny Wykłady: 8 Ćwiczenia: - Laboratoria: 8 Projekty/seminaria: -		Liczba punktów 3
Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny) (brak)		(ogólnouczelniany, z innego kierunku) (brak)
Obszar(y) kształcenia i dziedzina(y) nauki i sztuki nauki techniczne		Podział ECTS (liczba i %) 3 100%
Odpowiedzialny za przedmiot / wykładowca:		
dr inż. Grażyna Brzykcy email: Grażyna.Brzykcy@put.poznan.pl tel. 61 665 37 24 Wydział Elektryczny ul. Piotrowo 3A 60-965 Poznań		dr inż. Adam Meissner email: Adam.Meissner@put.poznan.pl tel. 61 665 37 24 Wydział Elektryczny ul. Piotrowo 3A 60-965 Poznań
Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:		
1	Wiedza:	Student ma podstawowe wiadomości z zakresu logiki matematycznej, teorii funkcji rekurencyjnych, programowania imperatywnego i deklaratywnego, programowania zorientowanego obiektowo, baz danych, systemów operacyjnych oraz sieci komputerowych.
2	Umiejętności:	Student potrafi pozyskiwać informacje z literatury, baz danych i innych źródeł; potrafi integrować uzyskane informacje, dokonywać ich interpretacji, a także wyciągać wnioski oraz formułować opinie; posługuje się językiem angielskim w stopniu wystarczającym do porozumiewania się, a także do czytania ze zrozumieniem wskazanej literatury przedmiotowej.
3	Kompetencje społeczne	Student rozumie potrzebę i zna możliwości ciągłego doksztalcania się, podnoszenia kompetencji językowych, zawodowych, osobistych i społecznych; ma świadomość odpowiedzialności za pracę własną oraz gotowość podporządkowania się zasadom pracy w zespole i ponoszenia odpowiedzialności za wspólnie realizowane zadania.
Cel przedmiotu: przegląd paradygmatów obliczeniowych wykorzystywanych w informatyce ze szczególnym uwzględnieniem podstawowych pojęć, technik i abstrakcji programowych; wypracowanie umiejętności doboru modelu obliczeniowego do rozwiązywanego problemu; nabycie umiejętności programowania w wieloparadygmatowym środowisku obliczeniowym.		
Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia		
Wiedza: 1. Student ma uporządkowaną i podbudowaną teoretycznie wiedzę w zakresie podst. konstrukcji programistycznych, implementacji algorytmów, paradygmatów i stylów programowania, metod weryfikacji poprawności programów, języków formalnych, kompilatorów, platform - [K_W05]		
Umiejętności: 1. Student potrafi posłużyć się środowiskami i platformami programistycznymi do pisania, wykonywania i testowania prostych programów kodowanych w językach programowania imperatywnego, obiektowego i deklaratywnego - [K_U10]		
Kompetencje społeczne: 1. Student ma świadomość ważności dokładnego wykonania projektu, zachowania standardów notacyjnych, przestrzegania poprawności językowej i terminowego oddania prac - [K_K07]		
Sposoby sprawdzenia efektów kształcenia		
Wykład. Pisemne sprawdzenie wiadomości obejmujące pytania teoretyczne oraz proste zadania. Laboratorium. Ustna lub pisemna ocena przygotowania studentów do zajęć. Ocena aktywności studentów w zakresie realizacji bieżących ćwiczeń. Warunki zaliczenia wykładu i laboratorium: należy uzyskać co najmniej 50,1% łącznej liczby punktów.		

Treści programowe		
<p>Wykład. Deklaratywny paradygmat obliczeniowy. Charakterystyczne pojęcia i techniki programowania funkcyjnego i deterministycznego programowanie w logice. Programowanie iteracyjne, rekurencyjne, metaprogramowanie, abstrakcyjne typy danych. Deklaratywna równoległość, czyli sterowany danymi model obliczeń równoległych. Model równoległości ze współdzielonym stanem. Relacyjny paradygmat obliczeniowy i bazy danych. Programowanie rozproszone i programowanie z więzami jako elementy metodyki programowania wieloparadygmatowego.</p> <p>Laboratorium. Ćwiczenia obejmujące techniki programowania wieloparadygmatowego w języku Oz z wykorzystaniem środowiska Mozart.</p>		
<p>Literatura podstawowa:</p> <ol style="list-style-type: none"> 1. Van Roy P., Haridi S.: Programowanie. Koncepcje, techniki i modele, Helion, Gliwice 2005. 2. Mozart Consortium: The Mozart programming system, http://www.mozart-oz.org, 2006. 		
<p>Literatura uzupełniająca:</p> <ol style="list-style-type: none"> 1. Brzykcy G., Meissner. A.: Programowanie w Prologu i programowanie funkcyjne. Materiały do ćwiczeń, Wyd. PP., 1999. 2. Kowalski R.: Logika w rozwiązywaniu zadań, WNT, Warszawa, 1989. 		
Bilans nakładu pracy przeciętnego studenta		
Czynność	Czas (godz.)	
1. Wykłady	8	
2. Laboratoria	8	
3. Przygotowanie do zajęć	48	
Obciążenie pracą studenta		
forma aktywności	godzin	ECTS
Łączny nakład pracy	64	3
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	16	1
Zajęcia o charakterze praktycznym	48	2