

<b>Nazwa przedmiotu</b> Wstęp do programowania PSPI		<b>Kod ECTS</b> 11.3.0056				
<b>Nazwa jednostki prowadzącej przedmiot</b> Instytut Informatyki						
<b>Nazwisko osoby prowadzącej (osób prowadzących)</b> dr inż. Jerzy Skurczyński						
<b>Studia</b>						
wydział	kierunek	stopień	tryb	specjalność	specjalizacja	semestr
Wydział Matematyki, Fizyki i Informatyki	Studia Podyplomowe Podstaw Informatyki	wszystkie	wszystkie	wszystkie	wszystkie	1
<b>Formy zajęć, sposób ich realizacji i przypisana im liczba godzin</b>				<b>Liczba punktów ECTS</b>		
<b>Formy zajęć</b> Wykład, Ćw. laboratoryjne				7		
<b>Sposób realizacji zajęć</b> zajęcia w sali dydaktycznej				20 godz wykładu, 30 godz ćwiczeń oraz praca własna studenta		
<b>Liczba godzin</b> Ćw. laboratoryjne: 30 godz., Wykład: 20 godz.						
<b>Cykl dydaktyczny</b> 2012/2013 zimowy						
<b>Status przedmiotu</b> obowiązkowy			<b>Język wykładowy</b> polski			
<b>Metody dydaktyczne</b> - wykład z prezentacją multimedialną - ćwiczenia laboratoryjne - projektowanie doświadczeń - ćwiczenia laboratoryjne - wykonywanie doświadczeń			<b>Forma i sposób zaliczenia oraz podstawowe kryteria oceny lub wymagania egzaminacyjne</b>			
			<b>Sposób zaliczenia</b> - Egzamin - Zaliczenie na ocenę			
			<b>Formy zaliczenia</b> - ustalenie oceny zaliczeniowej na podstawie ocen cząstkowych otrzymywanych w trakcie trwania semestru - egzamin pisemny (dłuższa wypowiedź pisemna / rozwiązanie problemu)			
			<b>Podstawowe kryteria oceny</b> Egzamin - punkty uzyskane za rozwiązanie zadań. Ćwiczenia - oceny uzyskiwane za projektowane i uruchamiane programy.			
<b>Określenie przedmiotów wprowadzających wraz z wymogami wstępnymi</b>						
<b>A. Wymagania formalne</b> Ukończona szkoła średnia.						
<b>B. Wymagania wstępne</b> Znajomość matematyki w zakresie szkoły średniej.						
<b>Cele kształcenia</b> Nauczenie studentów projektowania, tworzenia i uruchamiania prostych programów w wybranym środowisku programistycznym. Wyrobienie właściwych nawyków programistycznych.						
<b>Treści programowe</b> Cykl powstawania i uruchamiania programu. Pojęcie poprawności częściowej i całkowitej. Rodzaje błędów (składniowe, czasu wykonania, logiczne). Zapis prostych algorytmów przy użyciu: a) uporządkowanego opisu w języku naturalnym; b) schematów blokowych. Instrukcja przypisania i proste instrukcje sterujące w wybranym języku programowania. Przykłady prostych programów. Pojęcie funkcji i jej wywołanie w programie.						
<b>Wykaz literatury</b> 1. D. Harel, Rzecz o istocie informatyki (algorytmika), WNT, 2005. 2. B. Kernighan, D. Ritchie, Język ANSI C, WNT, 2004.						
<b>Efekty uczenia się</b> Student: Zna podstawowe konstrukcje programistyczne i struktury danych oraz metody projektowania algorytmów i programów Posiada umiejętność konstruowania rozumowań matematycznych Potrafi projektować, pisać, uruchamiać i testować programy			<b>Wiedza</b> Student zna zasób podstawowych konstrukcji algorytmicznych (warunki, pętle, wywołanie podprogramu) i jest w stanie wyrazić je w wybranym języku programowania. Ma świadomość znaczenia pojęcia poprawności programu.			
			<b>Umiejętności</b> Student umie posłużyć się podstawowymi narzędziami służącymi do tworzenia, uruchamiania i diagnostyki programów w wybranym języku (edytor, kompilator, narzędzia do śledzenia wykonywania programu). Umie przeprowadzać proste			

w wybranym środowisku programistycznym Rozumie i potrafi wyjaśnić niskopoziomowe zasady wykonywania programów	rozumowania prowadzące do wykrycia błędów w programach. <b>Kompetencje społeczne (postawy)</b> Student umie współpracować z zespołem programistów i wspólnie rozwiązywać napotkane problemy. Potrafi dzielić się wiedzą z innymi studentami i korzystać z ich wiedzy.
<b>Kontakt</b> jsk@inf.ug.edu.pl	