

Nazwa przedmiotu Programowanie 1		Kod ECTS		
Nazwa jednostki prowadzącej przedmiot Uniwersytet Opolski, Wydział Matematyki, Fizyki i Informatyki, Instytut matematyki i Informatyki				
Studia				
	Kierunek	stopień	tryb	specjalność
	Informatyka	Pierwszy	Stacjonarne Niestacjonarne *)	
Nazwisko osoby prowadzącej (osób prowadzących)				
Formy zajęć, sposób ich realizacji i przypisana im liczba godzin		Liczba punktów ECTS: 10		
A. Formy zajęć <ul style="list-style-type: none"> wykład (W); laboratorium (L); konwersatorium (K). 		Kalkulacja nakładu pracy: 45 godz. – uczestnictwo w wykładach [*]26]; 15 godz. – utrwalenie i pogłębienie wiedzy zakresie objętym wykładem [*]34]; 45 godz. – uczestnictwo w laboratoriach [*]26]; 90 godz. – dokończenie w domu zadań programistycznych [*]109]; 30 godz. – uczestnictwo w konwersatoriach [*]18]; 30 godz. – przygotowanie w domu rozwiązań zadań na konwersatorium [*]45]; 5 godz. – udział w konsultacjach [*]2].		
B. Sposób realizacji <ul style="list-style-type: none"> zajęcia w sali wykładowej/laboratoryjnej/dydaktycznej. 		Sumaryczny nakład pracy: 260 godzin, w tym		
C. Liczba godzin <ul style="list-style-type: none"> wykład – 45 godzin; laboratorium – 45 godzin; konwersatorium – 30 godzin. *) Studia niestacjonarne: <ul style="list-style-type: none"> wykład – 26 godzin (2T+24Z); laboratorium – 26 godzin; konwersatorium – 18 godzin. 		<ul style="list-style-type: none"> nakład pracy związany z zajęciami wymagającymi bezpośredniego udziału nauczycieli akademickich: 125 godz., co odpowiada 5 pkt. ECTS; nakład pracy związany z zajęciami o charakterze praktycznym: 195 godz., co odpowiada 7 pkt. ECTS. *) na studiach niestacjonarnych: <ul style="list-style-type: none"> nakład pracy związany z zajęciami wymagającymi bezpośredniego udziału nauczycieli akademickich: 72 godz., co odpowiada 3 pkt. ECTS; nakład pracy związany z zajęciami o charakterze praktycznym: 195 godz., co odpowiada 7 pkt. ECTS. 		
Status przedmiotu <ul style="list-style-type: none"> obowiązkowy. 		Język wykładowy polski		
Metody dydaktyczne <ul style="list-style-type: none"> wykład / wykład problemowy / wykład z prezentacją multimedialną; ćwiczenia laboratoryjne: wykorzystanie zintegrowanego środowiska programistycznego; konwersatorium: dyskusja, rozwiązywanie zadań. 		Forma i sposób zaliczenia oraz podst. kryteria oceny lub wymagania egzaminacyjne na ogólnych zasadach określonych w programie kształcenia, a w szczególności		
		A. Sposób zaliczenia <ul style="list-style-type: none"> (W, K, L) zaliczenie z oceną. 		
		B. Formy zaliczenia <ul style="list-style-type: none"> (W, K, L) ustalenie oceny zaliczeniowej na podstawie ocen cząstkowych otrzymywanych w trakcie trwania semestru za wystąpienia ustne i za prace pisemne, w tym programy komputerowe. 		
		C. Podstawowe kryteria <ul style="list-style-type: none"> (W, K, L) uzyskanie pozytywnej oceny zaliczeniowej. 		
Określenie przedmiotów wprowadzających wraz z wymogami wstępnymi brak wymagań wstępnych				
Cele przedmiotu Celem jest zapoznanie studentów z pojęciami algorytmu i programu, nauczenie projektowania i zapisywania w języku imperatywnym prostych programów oraz uruchamiania ich w środowisku programistycznym, a także weryfikowania ich poprawności.				
Treści programowe				
A. Problematyka wykładu i konwersatorium: Programowanie niskopoziomowe – maszyna RAM, pojęcie algorytmu, znane algorytmy iteracyjne, implementacja algorytmów z wykorzystaniem podstawowych konstrukcji programistycznych w języku niskiego poziomu, dynamiczny przydział pamięci, paradygmat programowania imperatywnego, podstawowe struktury danych i wykonywane na nich operacje, podstawowe konstrukcje programistyczne w języku wysokiego poziomu, w tym instrukcje warunkowe i iteracyjne, funkcje i procedury, zakres działania zmiennych, rekurencja i jej implementacja, metody weryfikacji poprawności programów, typ wskaźnikowy i referencyjny, alokacja i zwalnianie pamięci dynamicznej.				

B. Problematyka laboratorium:

Praktyczne wykorzystanie wiedzy nabytej na wykładzie w celu zapisywania algorytmów w języku programowania niskiego i wysokiego poziomu oraz kompilowania, uruchamiania i testowania programów komputerowych w zintegrowanym środowisku programistycznym.

Wykaz literatury

1. N. Wirth, Algorytmy + struktury danych = programy, Wyd. Naukowo-Techniczne 2004.
2. Matthias Felleisen, Robert Bruce Findler, Matthew Flatt, Shriram Krishnamurthi, Projektowanie oprogramowania. Wstęp do programowania i techniki komputerowej, Helion 2003.
3. Edsger W. Dijkstra, Umiejętność programowania, Wyd. Naukowo-Techniczne 1985.

Efekty kształcenia	Wiedza			
	Symb.	Efekt	Metoda weryfikacji	Odniesienie
	W01	Zna pojęcie algorytmu	Sprawdziany pisemne	K_W03, K_W04
	W02	Zna podstawowe konstrukcje programistyczne		K_W03, K_W04
	W03	Zna podstawowe struktury danych i wykonywane na nich operacje		K_W03, K_W04
	W04	Zna wskaźniki i referencje oraz dynamiczny przydział pamięci		K_W03, K_W04
	W05	Zna rekurencję w programowaniu imperatywnym		K_W03, K_W04
	W06	Zna metody weryfikacji poprawności programów		K_W03, K_W04
	Umiejętności:			
	Symb.	Efekt	Metoda weryfikacji	Odniesienie
	U01	Potrafi zapisać proste algorytmy w postaci schematu blokowego	Sprawdzian pisemny	
	U02	Potrafi pisać, uruchamiać i testować proste programy wykorzystujące instrukcje warunkowe i konstrukcje pętli w wybranym środowisku programistycznym	Zadania programistyczne	K_U09
	U03	Potrafi pisać, uruchamiać i testować programy zawierające własne procedury/funkcje w wybranym środowisku programistycznym		K_U09
	U04	Potrafi pisać, uruchamiać i testować programy wykorzystujące procedury/funkcje rekurencyjne w wybranym środowisku programistycznym		K_U09
	U05	Potrafi czytać ze zrozumieniem programy zapisane w języku programowania imperatywnego		K_U10
	U06	Umie pisać proste programy w kodzie maszyny RAM		K_U12
	Kompetencje społeczne (postawy)			
	Symb.	Efekt	Metoda weryfikacji	Odniesienie
K01	Potrafi wykonywać proste zadania w małym zespole	Zespołowe zadanie programistyczne	K_K02	
K02	Rozumie potrzebę pogłębiania swej wiedzy w zakresie programowania	Praca pisemna lub konwersacja	K_K05	

Kontakt:

Wykaz numerów telefonicznych i adresów mailowych pracowników znajduje się na stronie Instytutu Matematyki i Informatyki:
www.math.uni.opole.pl