

# SYLABUS ZAJ /GRUPY ZAJ

## Dane ogólne:

Jednostka organizacyjna:	Katedra Automatyki i Robotyki				
Kierunek studiów:	Automatyka i robotyka				
Specjalno /Specjalizacja:					
Nazwa zaj / grupy zaj :	J zyki i techniki programowania I				
Course / group of courses:	Languages and Techniques of Programming I				
Forma studiów:	stacjonarne				
Nazwa katalogu:	WP-AR-I-21/22Z				
Nazwa bloku zaj :					
Kod zaj /grupy zaj :	148609	Kod Erasmus:			
Punkty ECTS:	7	Rodzaj zaj :		obowi zkowy	
Rok studiów:	1	Semestr:		1	
Rok	Semestr	Forma zaj	Liczba godzin	Forma zaliczenia	ECTS
1	1	P	30	Zaliczenie z ocen	2.5
		LO	30	Zaliczenie z ocen	2.5
		W	30	Zaliczenie z ocen	2
Razem			90		7
Koordynator:		prof. dr hab. in . Jan Duda			
Prowadz cy zaj cia:					
J zyk wykładowy:		semestr: 1 - j zyk polski			

## Obja nienia:

Rodzaj zaj : obowi zkowe, do wyboru.

Forma prowadzenia zaj : W - wykład, - wiczenia audytoryjne, L - lektorat, S – seminarium/ zaj cia seminaryjne, P - wiczenia praktyczne (w tym zaj cia wf), M - wiczenia specjalistyczne (medyczne/ kliniczne), LO – wiczenia laboratoryjne, LI - laboratorium informatyczne, ZTI - zaj cia z technologii informacyjnych, P – wiczenia projektowe, ZT – zaj cia terenowe, T - wiczenia terenowe na obozach programowych, SK - samokształcenie (i inne), PR - praktyka zawodowa

## Dane merytoryczne

Wymagania wst pne:			
Znajomo zasad korzystania z komputera. Znajomo podstaw matematyki na poziomie szkoły redniej, umiej tno logicznego i kreatywnego my lenia.			
Szczegółowe efekty uczenia si			
Lp.	Student, który zaliczył zaj cia, zna i rozumie/potrafi/jest gotowy do:	Kod efektu dla kierunku studiów	Sposób weryfikacji efektu uczenia si
1	Ma uporz dkowan wiedz n/t zasad algorytmizacji zada i cyfrowego kodowania algorytmów.	AR1_W03	kolokwium
2	Zna zasady ogólne programowania strukturalnego, proceduralnego i obiektowego oraz budowania oprogramowania z wykorzystaniem ró nych j zyków programowania, zna zasady doboru j zyka programowania do specyfiki zadania programistycznego, zna i rozumie zasady niezawodnego programowania komputerów.	AR1_W05	kolokwium
3	Umie stosowa składni i semantyk j zyka C (w tym arytmetyk wska ników) dla budowania prostego niezawodnego oprogramowania w tym j zyku.	AR1_U03	kolokwium

4	Potrafi zaprojektować strukturę oprogramowania, potrafi zbudować w języku C niezawodny prosty program obliczeniowy z wykorzystaniem arytmetyki wskaźnikowej, wprowadza dane z klawiatury i plików oraz przekazuje wyniki na standardowe urządzenia zewnętrzne (monitor, pliki dyskowe).	AR1_U12	kolokwium, ocena aktywności
<b>Stosowane metody osiągnięcia zakładanych efektów uczenia się (metody dydaktyczne)</b>			
metody praktyczne (ćwiczenia praktyczne z odpytywaniem (sprawdzanie wiedzy i umiejętności).), metody praktyczne (Laboratorium - sprawdzanie umiejętności samodzielnego rozwiązywania problemów programistycznych.), metody podające (Wykład z elementami aktywizującymi (odpytywanie, zachęcanie do dyskusji różnych rozwiązań algorytmicznych).)			
<b>Kryteria oceny i weryfikacji efektów uczenia się</b>			
<b>wiedza:</b> ocena kolokwium (Sprawdziany na ćwiczeniach i laboratorium.) <b>umiejętności:</b> ocena kolokwium (Sprawdziany na ćwiczeniach i laboratorium.) ocena aktywności (Ocena aktywności na zajęciach)			
<b>Warunki zaliczenia</b>			
Wykład: testy, ćwiczenia praktyczne, ćwiczenia laboratoryjne: Oceny z kolokwium. Do zaliczenia przedmiotu ocena z ćwiczeń musi być pozytywna. Prowadzenie listy obecności na wykładach.			
<b>Treści programowe (opis skrócony)</b>			
Zasady konstruowania i kodowania algorytmów obliczeniowych. Ogólne zasady niezawodnego programowania. Środowiska programistyczne oraz zasady uruchamiania i testowania oprogramowania (diagnostyka i testowanie - wykorzystanie debuggerów). Szczegółowe zasady programowania w języku C (z odniesieniami do innych języków), rola preprocesingu, zasady arytmetyki wskaźnikowej, gospodarka pamięcią, instrukcje arytmetyczne logiczne, sterujące, biblioteki.			
<b>Content of the study programme (short version)</b>			
The principles of constructing and coding computational algorithms. General principles of reliable programming. Programming environments and rules for running and testing software (diagnostics and testing - the use of debuggers). Detailed rules of programming in the C language (with references to other languages), the role of preprocessing, the principles of indicator arithmetic, memory management, logical and control arithmetic instructions, libraries.			
<b>Treści programowe</b>			
			Liczba godzin
Semestr: 1			
Forma zajęć : <b>wykład</b>			
Zasady bitowego i cyfrowego kodowania informacji, typy danych, rozkazy, dane, rejestry, pamięć, urządzenia zewnętrzne. Algorytmy i ich schematy blokowe. Zasady komputerowego przetwarzania informacji. Zasady kodowania algorytmów - konstrukcja programu (nazwy, słowa kluczowe, operatory). Interpreterzy i kompilatory, pliki źródłowe, binarne i wykonywalne. Edycja wersji źródłowej, kompilacja i ładowanie – rola stylu programowania, diagnostyka poprawności syntaktycznej. Zasady testowania oprogramowania. Zasady programowania w języku C: struktura programu (pliki źródłowe, moduły, funkcje, biblioteki); struktura modułu (deklaracje, bloki, instrukcje, zasięg globalności nazw, komentarze). Deklaracje obiektów języka C (struktura instrukcji deklarujących i ich miejsce w kodzie). Podstawowe operacje preprocesora (rola plików nagłówkowych i ich doładowanie, stałe symboliczne). Obiekty języka C: stałe, zmienne proste, tablice, łańcuchy znaków, funkcje. Zmienne wskaźnikowe, operacje na wskaźnikach, wskaźniki a tablice. Rzutowanie typu, typy definiowane, rozmiar obiektu. Operatory i kolejno wykonywania operacji. Konstrukcje algorytmów w języku C: instrukcje arytmetyczne, instrukcje sterujące, pętle. Operacje wejścia i wyjścia: funkcje czytania znaków i łańcuchów znakowych, specyfikacje formatu. Zasady niezawodnego programowania.			30
Forma zajęć : <b>wiczenia praktyczne</b>			
Realizacja treści wykładu na ćwiczeniach praktycznych.			30
Forma zajęć : <b>wiczenia laboratoryjne</b>			
Realizacja treści wykładu na ćwiczeniach laboratoryjnych.			30
<b>Literatura</b>			
Podstawowa			
B.W.Kernighan, D.M.Ritchie, Język C, WNT, Warszawa 1992			

K.A.Barklay, ANSI C – Problem Solving an Programming, Printice Hall 1990
N. Wirth, Algorytmy + struktury danych = programy, WNT, Warszawa 2002
Uzupełniaj ca
D. van Tassel, Praktyka programowania, WNT, Warszawa 1989
G. Myers, Projektowanie niezawodnego oprogramowania, WNT, Warszawa 1989
W.Duch, Fascynuj cy wiat komputerów, Nakom, Pozna 1997

#### Dane jako ciowe

Przyporzkowanie zaj /grup zaj do dyscypliny naukowej/artystycznej		informatyka techniczna i telekomunikacja	
Sposób okre lenia liczby punktów ECTS			
Forma nakładu pracy studenta (udział w zaj ciach, aktywno , przygotowanie sprawozdania, itp.)		Obci enia studenta [w godz.]	
Udział w zaj ciach		90	
Konsultacje z prowadz cym		2	
Udział w egzaminie		0	
Bezpo redni kontakt z nauczycielem - inne		18	
Przygotowanie do laboratorium, wicze , zaj		25	
Przygotowanie do kolokwiiów i egzaminu		25	
Indywidualna praca własna studenta z literatur , wykładami itp.		25	
Inne		25	
Sumaryczne obci enie prac studenta		210	
Liczba punktów ECTS			
Liczba punktów ECTS		7	
Zaj cia wymagaj ce bezpo redniego udziału nauczyciela akademickiego		L. godzin	ECTS
		110	3,7
Zaj cia o charakterze praktycznym		L. godzin	ECTS
		140	4,7

1 godz = 45 minut; 1 punkt ECTS = 25-30 godzin

W sekcji 'Liczba punktów ECTS' suma punktów ECTS zaj wymagaj cych bezpo redniego udziału nauczyciela akademickiego i o charakterze praktycznym mo e si ró nić od ł cznej liczby punktów ECTS dla zaj /grup zaj .