

# SYLABUS ZAJ /GRUPY ZAJ

## Dane ogólne:

Jednostka organizacyjna:	Katedra Informatyki				
Kierunek studiów:	Informatyka				
Specjalno /Specjalizacja:					
Nazwa zaj / grupy zaj :	Programowanie w C				
Course / group of courses:	Programming in C				
Forma studiów:	stacjonarne				
Nazwa katalogu:	WP-IN-I-20/21Z				
Nazwa bloku zaj :					
Kod zaj /grupy zaj :	105960	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
		LO	30	Zaliczenie z ocen	2
		W	30	Egzamin	3
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, PR - praktyka zawodowa

## Dane merytoryczne

Wymagania wst pne:			
Znajomo zasad korzystania z komputera. Znajomo podstaw matematyki na poziomie szkoły redniej, umie j 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	posiada zaawansowan wiedz w zakresie j zyków, metod, algorytmów oraz paradygmatów programowania, ma wiedz w zakresie modelowania, analizowania oraz przetwarzania danych	IN1_W07, IN1_W03	egzamin, ocena aktywno ci
2	zna i rozumie cykl ycia oprogramowania oraz etapy wytwarzania w zakresie projektowania, implementacji, testowania oraz wdrowania, ma wiedz w zakresie tworzenia oprogramowania dla ró nych zastosowa informatyki	IN1_W08	egzamin, ocena aktywno ci
3	potrafi sformułowa specyfikac prostych systemów informatycznych na poziomie realizowanych funkcji, tak e z wykorzystaniem standardowych notacji	IN1_U05	egzamin, wykonanie zadania, praca pisemna

4	potrafi konstruować, integrować oraz implementować algorytmy z wykorzystaniem podstawowych technik algorytmicznych, a także dokonać analizy złożoności obliczeniowej	IN1_U09	wykonanie zadania, praca pisemna
5	jest świadomy wartości, dostrzega i rozumie pozatechniczne i etyczne uwarunkowania działalności inżynierskiej	IN1_U10	wykonanie zadania, obserwacja zachowa
6	potrafi samodzielnie planować i realizować własne uczenie się przez całe życie, m. in. w celu podnoszenia kompetencji zawodowych	IN1_U14	praca pisemna, obserwacja zachowa
7	jest gotów do krytycznej oceny efektów swojej pracy oraz uznawania wiedzy w rozwiązywaniu problemów poznawczych i praktycznych oraz zasięgania opinii ekspertów w przypadku wystąpienia problemów	IN1_K01	egzamin, praca pisemna, obserwacja zachowa
8	przestrzega zasad etyki zawodowej, jest świadomy wartości zachowania w sposób profesjonalny	IN1_K05	egzamin, wykonanie zadania, obserwacja zachowa
<b>Stosowane metody osiągnięcia zakładanych efektów uczenia się (metody dydaktyczne)</b>			
metody podające (wykład tradycyjny (informacyjny) z demonstracją przykładów.), metody podające (objaśnienie (wyjaśnienie, omówienie).)			
<b>Kryteria oceny i weryfikacji efektów uczenia się</b>			
<b>wiedza:</b> egzamin (Egzamin ustny praktyczny) ocena aktywności (ocena aktywności na zajęciach (pytania/ odpowiedzi)) <b>umiejętności:</b> egzamin (Egzamin ustny praktyczny) obserwacja zachowa (obserwacja zachowa indywidualnych i zespołowych pod kątem kompetencji społecznych) ocena pracy pisemnej (ocena pracy zaliczeniowej) ocena wykonania zadania (ocena wykonania zadania indywidualnego lub zespołowego na ćwiczeniach, na laboratorium) <b>kompetencje społeczne:</b> egzamin (Egzamin ustny praktyczny) obserwacja zachowa (obserwacja zachowa indywidualnych i zespołowych pod kątem kompetencji społecznych) ocena pracy pisemnej (ocena pracy zaliczeniowej) ocena wykonania zadania (ocena wykonania zadania indywidualnego lub zespołowego na ćwiczeniach, na laboratorium)			
<b>Warunki zaliczenia</b>			
Uzyskanie zaliczenia z ćwiczeń praktycznych i laboratoryjnych, zdanie egzaminu ustnego praktycznego na podstawie kodu aplikacji w języku C przygotowanej samodzielnie przez studenta. 1. Stopień rozumienia istoty kodowania i przetwarzania informacji w komputerze; 2. Znajomość zasad funkcjonowania języków programowania 3. Biegła znajomość syntaktyki języka C i stopień rozumienia wpływu poszczególnych instrukcji na stan pamięci operacyjnej 4. Znajomość zasad podziału kodu na pliki i budowania modułów oprogramowania 5. Znajomość zasad niezawodnego programowania, rozumienie różnic budów oprogramowania i ich rangi 6. Stopień wykorzystania talentów programistycznych studenta			
<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 pamięci, instrukcje arytmetyczne logiczne, sterujące, biblioteki, aplikacje wielozadaniowe i komunikacja międzyzadaniowa.			
<b>Content of the study programme (short version)</b>			
Principles of constructing and coding computational algorithms. General principles of reliable programming. Programming tools 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 pointer arithmetic, memory management, logical and arithmetic instructions, libraries, multitask applications and interprocess communication.			
<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, uruchamianie zewnętrzne. Algorytmy i ich schematy blokowe. Zasady komputerowego przetwarzania informacji. Zasady kodowania algorytmów - konstrukcja programu (nazwy, słowa kluczowe, operatory). Interpretatory i kompilatory, pliki źródłowe, binarne i wykonywalne. Edycja wersji źródłowej, kompilacja i			30

<p>ł czenie – rola stylu programowania, diagnostyka poprawno ci syntaktycznej. Zasady testowania oprogramowania. Zasady programowania w j zyku C: struktura programu (pliki ródlowe, moduły, funkcje, biblioteki); struktura modułu (deklaracje, bloki, instrukcje, zasi g globalno ci nazw, komentarze). Deklaracje obiektów j zyka C (struktura instrukcji deklaruj cych i ich miejsce w kodzie). Podstawowe operacje preprocesora (rola plików nagłówkowych i ich doł czanie, stałe symboliczne). Obiekty j zyka C: stałe, zmienne proste, tablice, ła cuchy znaków, funkcje. Zmienne wska nikowe, operacje na wska nikach, wska niki a tablice. Rzutowanie typu, typy definiowane, rozmiar obiektu. Operatory i kolejno wykonywania operacji. Konstrukcje algorytmów w j zyku C: instrukcje arytmetyczne, instrukcje steruj ce, p tle. Operacje wej cia i wyj cia: funkcje czytania znaków i ła cuchów znakowych, specyfikacje formatu. Zasady niezawodnego programowania. Ogólne zasady budowania aplikacji wielozadaniowych i pracuj cych w re imie czasu rzeczywistego – komunikacja mi dzyprocesowa.</p>	30
Forma zaj : <b>wiczenia praktyczne</b>	
<p>Zasady bitowego i cyfrowego kodowania informacji, typy danych, rozkazy, dane, rejestry, pami , urz dzenia zewn trzne. Algorytmy i ich schematy blokowe. Zasady komputerowego przetwarzania informacji. Zasady kodowania algorytmów - konstrukcja programu (nazwy, słowa kluczowe, operatory). Interpretery i kompilatory, pliki ródlowe, binarne i wykonywalne. Edycja wersji ródlowej, kompilacja i ł czenie – rola stylu programowania, diagnostyka poprawno ci syntaktycznej. Zasady testowania oprogramowania. Zasady programowania w j zyku C: struktura programu (pliki ródlowe, moduły, funkcje, biblioteki); struktura modułu (deklaracje, bloki, instrukcje, zasi g globalno ci nazw, komentarze). Deklaracje obiektów j zyka C (struktura instrukcji deklaruj cych i ich miejsce w kodzie). Podstawowe operacje preprocesora (rola plików nagłówkowych i ich doł czanie, stałe symboliczne). Obiekty j zyka C: stałe, zmienne proste, tablice, ła cuchy znaków, funkcje. Zmienne wska nikowe, operacje na wska nikach, wska niki a tablice. Rzutowanie typu, typy definiowane, rozmiar obiektu. Operatory i kolejno wykonywania operacji. Konstrukcje algorytmów w j zyku C: instrukcje arytmetyczne, instrukcje steruj ce, p tle. Operacje wej cia i wyj cia: funkcje czytania znaków i ła cuchów znakowych, specyfikacje formatu. Zasady niezawodnego programowania. Ogólne zasady budowania aplikacji wielozadaniowych i pracuj cych w re imie czasu rzeczywistego – komunikacja mi dzyprocesowa.</p>	30
Forma zaj : <b>wiczenia laboratoryjne</b>	
<p>Zasady bitowego i cyfrowego kodowania informacji, typy danych, rozkazy, dane, rejestry, pami , urz dzenia zewn trzne. Algorytmy i ich schematy blokowe. Zasady komputerowego przetwarzania informacji. Zasady kodowania algorytmów - konstrukcja programu (nazwy, słowa kluczowe, operatory). Interpretery i kompilatory, pliki ródlowe, binarne i wykonywalne. Edycja wersji ródlowej, kompilacja i ł czenie – rola stylu programowania, diagnostyka poprawno ci syntaktycznej. Zasady testowania oprogramowania. Zasady programowania w j zyku C: struktura programu (pliki ródlowe, moduły, funkcje, biblioteki); struktura modułu (deklaracje, bloki, instrukcje, zasi g globalno ci nazw, komentarze). Deklaracje obiektów j zyka C (struktura instrukcji deklaruj cych i ich miejsce w kodzie). Podstawowe operacje preprocesora (rola plików nagłówkowych i ich doł czanie, stałe symboliczne). Obiekty j zyka C: stałe, zmienne proste, tablice, ła cuchy znaków, funkcje. Zmienne wska nikowe, operacje na wska nikach, wska niki a tablice. Rzutowanie typu, typy definiowane, rozmiar obiektu. Operatory i kolejno wykonywania operacji. Konstrukcje algorytmów w j zyku C: instrukcje arytmetyczne, instrukcje steruj ce, p tle. Operacje wej cia i wyj cia: funkcje czytania znaków i ła cuchów znakowych, specyfikacje formatu. Zasady niezawodnego programowania. Ogólne zasady budowania aplikacji wielozadaniowych i pracuj cych w re imie czasu rzeczywistego – komunikacja mi dzyprocesowa.</p>	30
<b>Literatura</b>	
Podstawowa	
B.W.Kernighan, D.M.Ritchie, J zyck 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

Dodatkowa

D. van Tassel, Praktyka programowania, WNT, Warszawa 1989

**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		0	
Udział w egzaminie		2	
Bezpo redni kontakt z nauczycielem - inne		3	
Przygotowanie do laboratorium, wicze , zaj		20	
Przygotowanie do kolokwiiów i egzaminu		30	
Indywidualna praca własna studenta z literatur , wykładami itp.		20	
Inne		10	
Sumaryczne obci enie prac studenta		175	
Liczba punktów ECTS			
Liczba punktów ECTS		7	
Zaj cia wymagaj ce bezpo redniego udziału nauczyciela akademickiego		L. godzin	ECTS
		95	3,8
Zaj cia o charakterze praktycznym		L. godzin	ECTS
		145	5,8

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 ni od ł cznej liczby punktów ECTS dla zaj /grup zaj .