

# KARTA PRZEDMIOTU

## 1. Informacje ogólne

<b>Nazwa przedmiotu i kod (wg planu studiów):</b>	Podstawy programowania i teoria informacji, C2
<b>Nazwa przedmiotu (j. ang.):</b>	Basics of programming and information theory
<b>Kierunek studiów:</b>	Informatyka
<b>Specjalność/specjalizacja:</b>	Sieciowe systemy informatyczne/Technologie internetowe i bazy danych/Informatyka praktyczna
<b>Poziom kształcenia:</b>	studia I stopnia
<b>Profil kształcenia:</b>	praktyczny (P)
<b>Forma studiów:</b>	studia stacjonarne / studia niestacjonarne
<b>Obszar kształcenia:</b>	nauki techniczne
<b>Dziedzina:</b>	nauki techniczne
<b>Dyscyplina nauki:</b>	informatyka
<b>Koordinator przedmiotu:</b>	dr Jolanta Wojtowicz

## 2. Ogólna charakterystyka przedmiotu

<b>Przynależność do modułu:</b>	kierunkowego
<b>Status przedmiotu:</b>	obowiązkowy
<b>Język wykładowy:</b>	polski
<b>Rok studiów, semestr:</b>	I, 1
<b>Forma i wymiar zajęć według planu studiów:</b>	stacjonarne - wykład 30 h, ćw. laboratoryjne 30 h, ćw. audytoryjne 15h niestacjonarne - wykład 30 h, laboratoryjne 15 h, ćw. audytoryjne 15h
<b>Interesariusze i instytucje partnerskie (nieobowiązkowe)</b>	
<b>Wymagania wstępne / Przedmioty wprowadzające:</b>	- Umiejętność posługiwania się komputerem, - Umiejętność logicznego myślenia, - Znajomość matematyki na poziomie szkoły średniej.

### 3. Bilans punktów ECTS

Całkowita liczba punktów ECTS (wg planu studiów; 1 punkt = 25-30 godzin pracy studenta, w tym praca na zajęciach i poza zajęciami): (A + B)	7	stacjonarne	Niestacjonarne
		<b>A. Liczba godzin wymagających bezpośredniego udziału nauczyciela (kontakto- wych, w czasie rzeczywistym, w tym testy, egzaminu etc) z podziałem na typy zajęć oraz całkowita liczba punktów ECTS osią- ganych na tych zajęciach</b>	obecność na wykładach obecność na ćwiczeniach laboratoryj- nych ćwiczenia audytoryjne udział w konsultacjach udział w egzaminie
<b>B. Poszczególne typy zadań do samokształ- cenia studenta (niewymagających bezpo- średniego udziału nauczyciela) wraz z pla- nowaną średnią liczbą godzin na każde i sumaryczną liczbą ECTS (np. praca w bi- bliotece, w sieci, na platformie e- learningowej, w laboratorium, praca nad projektem końcowym, przygotowanie ogólne; suma poszczególnych godzin powinna zga- dzać się z liczbą ogólną)</b>	przygotowanie do zajęć praca na platformie e-learningowej przygotowanie do egzaminu studiowanie zalecanej literatury	45 10 30 20 <b>105</b> 4	40 30 35 25 <b>130</b> 5
<b>C. Liczba godzin praktycz- nych/laboratoryjnych w ramach przedmio- tu oraz związana z tym liczba punktów ECTS (ta liczba nie musi być powiązana z liczbą godzin kontaktowych, niektóre zajęcia praktyczne/laboratoryjne mogą odbywać się bez udziału nauczyciela):</b>	udział w zajęciach praca samodzielna  <b>w sumie:</b> ECTS	30 15  45 1,6	15 30  45 1,6

### 4. Opis przedmiotu

<p><b>Cel przedmiotu:</b></p> <p>Zapoznanie z podstawami programowania, obejmującymi m.in. zasady formułowania i algorytmizacji zadań, sposoby zapisu algorytmu, etapy powstawania programu oraz terminologię programistyczną. Znajomość podstawowych pojęć z zakresu teorii informacji oraz optymalnych metod kodowania informacji. Stworzenie fundamentów dla zaawansowanych przedmiotów kierunkowych.</p>
<p><b>Metody dydaktyczne:</b> wykład - pokaz, laboratorium - wykonywanie doświadczeń, projektowanie doświadczeń, ćwiczenia- zajęcia tablicowe.</p>
<p><b>Treści kształcenia (w rozbiciu na formę zajęć (jeśli są różne formy) i najlepiej w punktach):</b></p> <p><b>Wykłady:</b></p> <ol style="list-style-type: none"> <li>1. Podstawy algorytmiki. Pojęcie algorytmu. Algorytmy liniowe, rozgałęzione, cykliczne. Rekurencja. Sposoby zapisu algorytmu.</li> <li>2. Definicje pojęć: program, translator, kompilator, interpreter, moduł programowy. Programowanie algorytmiczne, strukturalne, obiektowe. Programowanie metodą wstępującą i zstępującą.</li> </ol>

Przegląd języków programowania. Języki kompilowane, języki częściowo kompilowane, języki interpretowane. Podział języków programowania ze względu na paradygmat programowania. Język C – wybrany język programowania.

3. Struktura programu w języku C/C++. Typy danych, operatory i wyrażenia. Operacje wejścia i wyjścia.
4. Instrukcje sterujące przepływem danych w programie: if, if-else, switch. Zagnieżdżanie.
5. Iteracyjne instrukcje sterujące przepływem danych w programie: while, do-while, for. Pętle zagnieżdżone. Instrukcje break i continue.
6. Funkcje. Zmienne lokalne i globalne. Deklaracja funkcji. Zwracanie rezultatu przez funkcję. Przekazywanie zmiennych do funkcji.
7. Łańcuchy. Tablice jedno i wielowymiarowe.
8. Wskaźniki. Wskaźniki do tablic, wskaźniki do funkcji, wskaźniki do wskaźników. Tablice wskaźników. Arytmetyka wskaźników.
9. Pliki. Deklarowanie, otwieranie i zamykanie. Dodawanie danych do pliku.
10. Struktury: Definiowanie struktur. Operacje na strukturach.
11. Dynamiczne zarządzanie pamięcią. Operatory new i delete.
12. Sygnał jako nośnik informacji. Przetwarzanie analogowe i cyfrowe. Konwersja sygnałów - próbkowanie, dyskretyzacja, rastrowanie, kwantowanie.
13. Układy liczbowe. Kody dziesiętne, dwójkowe, heksadecymalne, znakowe. Algorytmy przeliczania liczb w układach: dziesiętnym, dwójkowym, ósemkowym i szesnastkowym. Realizacja podstawowych działań arytmetycznych i logicznych. Kod Gray'a. Kod bezpośredni.
14. Kodowanie w kanale bezszumowym. Podstawy teorii kodowania. Optymalne kodowanie. Kod Fano. Kody Shannona i Huffmana. Entropia źródła informacji. Twierdzenie Shannona o kodowaniu. Redundancja kodu.
15. Kanał komunikacyjny. Przepustowość kanału komunikacyjnego. Reguły decyzyjne. Poprawa wiarygodności kanału. Odległość Hamminga dla kodów. Twierdzenie o korygowaniu zakłóceń. Zabezpieczanie przed zakłóceniem za pomocą bitu parzystości.

### **Ćwiczenia laboratoryjne:**

1. Środowisko programistyczne Bloodshed Dev-C++ – instalacja i konfiguracja.
2. Struktura programu w języku C/C++. Typy danych, operatory i wyrażenia. Operacje wejścia i wyjścia. Instrukcje sterujące przepływem danych w programie: if, if-else, switch. Zagnieżdżanie.
3. Iteracyjne instrukcje sterujące przepływem danych w programie: while, do-while, for. Pętle zagnieżdżone. Instrukcje break i continue.
4. Funkcje. Zmienne lokalne i globalne. Deklaracja funkcji. Zwracanie rezultatu przez funkcję. Przekazywanie zmiennych do funkcji.
5. Łańcuchy. Tablice jedno i wielowymiarowe.
6. Wskaźniki. Wskaźniki do tablic, wskaźniki do funkcji, wskaźniki do wskaźników. Tablice wskaźników. Arytmetyka wskaźników.
7. Pliki. Deklarowanie, otwieranie i zamykanie. Dodawanie danych do pliku.
8. Struktury: Definiowanie struktur. Operacje na strukturach.
9. Dynamiczne zarządzanie pamięcią. Operatory new i delete.

**Ćwiczenia audytoryjne:**

1. Podstawy algorytmiki. Pojęcie algorytmu. Algorytmy liniowe, rozgałęzione, cykliczne. Sposoby zapisu algorytmu. Rekurencja. Algorytmiczne rozwiązywanie zadań.
2. Układy liczbowe. Kody dziesiętne, dwójkowe, heksadecymalne, znakowe. Algorytmy przeliczania liczb w układach: dziesiętnym, dwójkowym, ósemkowym i szesnastkowym. Realizacja podstawowych działań arytmetycznych i logicznych. Kod Gray'a. Kod bezpośredni.
3. Podstawy teorii kodowania. Optymalne kodowanie. Kod Fano. Kody Shannona i Huffmana. Entropia źródła informacji. Twierdzenie Shannona o kodowaniu. Redundancja kodu.
4. Przepustowość kanału komunikacyjnego. Reguły decyzyjne. Poprawa wiarygodności kanału. Odległość Hamminga dla kodów. Twierdzenie o korygowaniu zakłóceń. Zabezpieczanie przed zakłóceniem za pomocą bitu parzystości.

**Efekty kształcenia**

<b>Efekt przedmiotu</b>	<b>Student, który zaliczył przedmiot (spełnił minimum wymagań)</b>	<b>Efekt kierunkowy</b>
C2_W01 C2_W02 C2_W03	<b>Wiedza:</b> <ol style="list-style-type: none"> <li>1. Student zna i rozumie podstawowe zagadnienia algorytmiki: pojęcie algorytmu, cechy poprawnego algorytmu, sposoby przedstawiania algorytmów, złożoność algorytmów.</li> <li>2. Student zna podstawowe struktury danych, konstrukcje programistyczne oraz metody weryfikacji poprawności programów.</li> <li>3. Student zna podstawowe działania arytmetyczne oraz logiczne wykonywane na liczbach w systemie, dwójkowym, ósemkowym i szesnastkowym oraz posiada uporządkowaną wiedzę z zakresu teorii informacji Shannona i metod optymalnego kodowania.</li> </ol>	K_W06  K_W08  K_W15
C2_U01  C2_U02	<b>Umiejętności</b> <ol style="list-style-type: none"> <li>1. Potrafi czytać ze zrozumieniem i konstruować algorytmy rozwiązujące wybrane problemy matematyczne i zapisywać je w postaci kodu źródłowego języka C oraz w postaciach takich jak: pseudokod, schemat blokowy, lista kroków.</li> <li>2. Potrafi praktycznie wykorzystywać zdobytą wiedzę z zakresu teorii informacji i metod optymalnego kodowania.</li> </ol>	K_U10  K_U21
C2_K01  C2_K02	<b>Kompetencje społeczne</b> <ol style="list-style-type: none"> <li>1. Student rozumie potrzebę poznawania nowych narzędzi programistycznych wykorzystywanych w programowaniu wybranych problemów rzeczywistych.</li> <li>2. Rozumie potrzebę wsparcia analizę modeli matematycznych stosownymi narzędziami informatycznymi.</li> </ol>	K_K01  K_K08

<b>Sposoby weryfikacji efektów kształcenia:</b>				
<b>Lp.</b>	<b>Efekt przedmiotu</b>	<b>Sposób weryfikacji</b>	<b>Ocena formująca</b>	<b>Ocena końcowa</b>
1	C2_W01 C2_W02 C2_W02 C2_U02	Udokumentowana ocenami aktywność studenta na zajęciach. Sprawdzian pisemny ze znajomości podstaw algorytmiki. Sprawdzian pisemny ze znajomości systemów liczbowy, operacji arytmetycznych na nich wykonywanych oraz z teorii informacji Shannona i metod kodowania informacji.	oceny za aktywność, oceny z sprawdzinów pisemnych, egzamin	Średnia ocen formujących
2.	C2_U01	Udokumentowana ocenami aktywność studenta na zajęciach. Sprawdziany z umiejętności kodowania.	oceny za aktywność, oceny z sprawdzinów pisemnych.	Średnia ocen formujących
3	C2_K01 C2_K02	Obserwacja, pogadanka.	Oceny za aktywność	Średnia ocen formujących

<b>Kryteria oceny</b>		
	<b>w zakresie wiedzy</b>	<b>Efekt kształcenia</b>
Na ocenę 3,0	<p>Student uzyskał min. 50% wymaganej wiedzy w zakresie obowiązującego materiału. Student zna:</p> <ul style="list-style-type: none"> <li>- podstawowe pojęcia algorytmiki: pojęcie algorytmu, cechy poprawnego algorytmu, sposoby przedstawiania algorytmów, klasy złożoności algorytmów,</li> <li>- podstawowe struktury danych, konstrukcje programistyczne oraz metody weryfikacji poprawności programów,</li> <li>- podstawowe działania arytmetyczne oraz logiczne wykonywane na liczbach w systemie, dwójkowym, ósemkowym i szesnastkowym oraz posiada uporządkowaną wiedzę z zakresu teorii informacji Shannona i metod optymalnego kodowania.</li> </ul>	C2_W01  C2_W02  C2_W03
Na ocenę 5,0	<p>Student zdobył powyżej 95% wymaganej wiedzy w zakresie obowiązującego materiału. Student zna:</p> <ul style="list-style-type: none"> <li>- pojęcie algorytmu, cechy poprawnego algorytmu, sposoby przedstawiania algorytmów, klasy złożoności algorytmów i potrafi skonstruować algorytm rozwiązujący konkretne zadania oraz potrafi ocenić poprawność algorytmu i wybrać algorytm dający rozwiązanie optymalne</li> <li>- podstawowe konstrukcje programistyczne oraz metody weryfikacji poprawności programów i potrafi tą wiedzę wykorzystać do rozwiązywania wybranych zadań matematycznych. Ponadto potrafi przedstawić rozwiązania alternatywne i wskazać optymalne. Posiada wiedzę pozwalającą mu wykazać za-</li> </ul>	C2_W01  C2_W02

	<p>leżność pomiędzy rozmiarem danych a efektywnością programu.</p> <ul style="list-style-type: none"> <li>- układy liczbowe i algorytmy przeliczania liczb w układach: dziesiętnym, dwójkowym, ósemkowym i szesnastkowym. Posiada wiedzę, która umożliwia mu realizację podstawowych działań arytmetycznych i logicznych. Ponadto zna własności i zastosowanie poszczególnych kodów i systemów liczbowych w technice cyfrowej. Potrafi wskazać praktyczne zastosowanie optymalnych metod kodowania i teorii informacji Schannona</li> </ul>	C2_W03
<b>w zakresie umiejętności</b>		
Na ocenę 3,0	<p>Student uzyskał min. 50% wymaganych umiejętności w zakresie obowiązującego materiału. Student potrafi:</p> <ul style="list-style-type: none"> <li>- czytać ze zrozumieniem i konstruować algorytmy rozwiązujące wybrane problemy matematyczne i zapisywać je w postaci kodu źródłowego języka C oraz w postaciach takich jak: pseudokod, schemat blokowy, lista kroków. Potrafi modyfikować te rozwiązania i przedstawiać rozwiązania alternatywne.</li> <li>- praktycznie wykorzystywać zdobytą wiedzę z zakresu teorii informacji i metod optymalnego kodowania.</li> </ul>	C2_U01  C2_U02
Na ocenę 5,0	<p>Student uzyskał powyżej 95% umiejętności w zakresie obowiązującego materiału. Student umie:</p> <ul style="list-style-type: none"> <li>- czytać ze zrozumieniem i konstruować algorytmy rozwiązujące wybrane problemy matematyczne i zapisywać je w postaci kodu źródłowego języka C oraz w postaciach takich jak: pseudokod, schemat blokowy, lista kroków. Potrafi podać kilka algorytmów rozwiązujących dany problem i spośród nich wybrać algorytm optymalny.</li> <li>- potrafi wykorzystać poznane metody, w razie potrzeby odpowiednio je modyfikując do rozwiązywania problemów z zakresu informatyki i elektroniki.</li> </ul>	C2_U01  C2_U02
<b>w zakresie kompetencji społecznych</b>		
Na ocenę 3,0	Student osiągną wymagane kompetencje społeczne na poziomie min. 50%.	C2_K01 C2_K02
Na ocenę 5,0	Student osiągną wymagane kompetencje społeczne na poziomie wyższym niż 90%.	C2_K01 C2_K02

<b>Zalecana literatura</b>	
<b>Literatura podstawowa:</b>	<ol style="list-style-type: none"> <li>1. Perry, Greg M.: Język C w przykładach, Warszawa : "Mikom" , 2000</li> <li>2. Steve Oualline.: Język C. Programowanie. Helion 2003.</li> <li>3. Sysło Maciej M.: Algorytmy, Warszawa : Wydawnictwa Szkolne i Pedagog , 1997.</li> <li>4. Chojcan J., Rutkowski J., Zbiór zadań z teorii informacji i kodowania, Gliwice, Wyd. Politechniki Śląskiej 2001</li> </ol>
<b>Literatura uzupełniająca:</b>	<ol style="list-style-type: none"> <li>1. Niewierowicz T.: Świat algorytmów, Warszawa, Nasza Księgarnia, 1980</li> <li>3. Bauer F. L. , Goos G.: Informatyka, Warszawa, WNT, 1977</li> <li>4. Bauer F. L., Gnatz R., Hill U.: Zbiór zadań z informatyki, Warszawa, WNT, 1981</li> <li>5. Dąbrowski A.: O teorii informacji, Warszawa, WN i P, 1974.7.</li> <li>6. Łukasik Z.: Teoria informacji i bezpieczeństwo transmisji, Radom</li> <li>7. Prata S. C Primer Plus, Fifth Edition</li> </ol>

### ***Informacje dodatkowe:***

<b>Dodatkowe obowiązki prowadzącego wraz z szacowaną całkowitą liczbą godzin:</b> <i>(np. indywidualne konsultacje, poprawa prac, przygotowanie projektu zaliczeniowego, egzaminu, przygotowanie ćwiczeń e-learningowych). Przykład poniżej</i>
Przygotowanie i aktualizacja wykładów, ćwiczeń, egzaminu.– 45 godzin
poprawa sprawdzianów pisemnych – 15 godzin
Poprawa egzaminu - 10
Konsultacje – 10 godzin
<b>W sumie: 80 godzin</b>