

# I. KARTA PRZEDMIOTU

1. Nazwa przedmiotu: **PODSTAWY PROGRAMOWANIA**
2. Kod przedmiotu: **Ovi1**
3. Jednostka prowadząca: **Wydział Mechaniczno-Elektryczny**
4. Kierunek: **Mechatronika**
5. Specjalność: **Techniki Komputerowe w Mechatronice**
6. Moduł: **treści podstawowych**
7. Poziom studiów: **I stopnia**
8. Forma studiów: **niestacjonarne**
9. Semestr studiów: **I**
10. Profil: **praktyczny**
11. Prowadzący: **dr hab. inż. Jerzy Garus**

## CEL PRZEDMIOTU

<b>C1</b>	Zapoznanie z metodami opisu algorytmu dla potrzeb programowania w języku C.
<b>C2</b>	Zapoznanie z zasadami programowania w języku C.
<b>C3</b>	Nabywanie umiejętności programowania w języku C
<b>C4</b>	Nabywanie umiejętności rozwiązywania podstawowych problemów inżynierskich w języku C

## WYMAGANIA WSTĘPNE W ZAKRESIE WIEDZY, UMIEJĘTNOŚCI I INNYCH KOMPETENCJI

<b>1</b>	Wiedza z zakresu matematyki i technologii informacyjnej.
<b>2</b>	Znajomość pozycyjnych systemów zapisu liczb.

## EFEKTY KSZTAŁCENIA

<b>EK1</b>	Student zna zasady tworzenia oprogramowania w języku C. Rozróżnia pojęcia kompilacja, interpretacja konsolidacja. Potrafi zdefiniować pojęcie algorytm. Zna metody opisu algorytmu. Zna zasady budowy schematu blokowego. Potrafi omówić podstawowe struktury programistyczne. Potrafi napisać i uruchomić prosty program w języku C.
<b>EK2</b>	Student potrafi omówić instrukcje złożone w instrukcjach sterujących. Potrafi wyjaśnić działanie instrukcji warunkowej if ... else, pętli while i for, wyboru (switch) oraz zasad przerywania i kontynuacji wykonywania pętli. Zna operatory matematyczne, relacyjne, logiczne, bitowe i operator przypisania. Zna priorytety operatorów.
<b>EK3</b>	Student potrafi wymienić złożone typy danych. Zna zasady tworzenia tablic liczbowych i znakowych. Potrafi inicjalizować tablice. Zna zasady konwersji łańcuchów do liczb. Zna przeznaczenie i sposób deklaracji struktury.
<b>EK4</b>	Student zna zasady wprowadzania i wyprowadzania danych. Potrafi sterować formatem. Zna i potrafi zastosować praktycznie podstawowe funkcje z biblioteki STDIO. Zna zasady obsługi plików oraz plikowe funkcje wejścia/wyjścia.
<b>EK5</b>	Student rozumie pojęcie funkcja. Zna zasady przekazywania parametrów do funkcji i zwracania wyniku funkcji. Wie co to prototyp funkcji (typ funkcyjny) oraz czas życia i zakres ważności nazwy. Potrafi praktycznie korzystać z funkcji bibliotecznych.
<b>EK6</b>	Student zna przeznaczenie wskaźników. Zna działania na wskaźnikach. Rozumie pojęcie dynamiczna alokacja pamięć. Potrafi zastosować wskaźniki do przekazywania parametrów oraz dostępu do tablic i składników struktury. Potrafi napisać i uruchomić złożony program w języku C.

## TREŚCI PROGRAMOWE

	WYKŁADY	Liczba godzin
<b>W1</b>	Wprowadzenie do programowania w języku C. Algorytmy. Podstawowe struktury programistyczne. Tworzenie oprogramowania w Języku C: alfabet, składnia i semantyka, kompilacja i konsolidacja programu; zmienne i typy.	<b>2</b>

<b>W2</b>	Zmienne i wyrażenia. Operatory matematyczne, relacyjne, logiczne, bitowe i operator przypisania. Sterowanie przebiegiem programu: instrukcja if ... else; pętle while i for; instrukcja skoku; instrukcja switch.	<b>2</b>
<b>W3</b>	Typy danych: całkowite, znakowe i zmiennoprzecinkowe. Konwersja typów. Złożone typy danych: tablice liczbowe i znakowe, rekordy i unie.	<b>1</b>
<b>W4</b>	Operacje wejścia/wyjścia: operacje wejścia/wyjścia (biblioteka stdio); sterowanie formatem, operacje wejścia/wyjścia na plikach, obsługa błędów.	<b>1</b>
<b>W5</b>	Pojęcie funkcji, zwracanie wyniku, przekazywanie parametrów, prototyp funkcji, zmienne lokalne i globalne, funkcje biblioteczne.	<b>2</b>
<b>W6</b>	Zarządzanie pamięcią: wskaźniki, dynamiczna alokacja pamięci, operator sizeof(), wskaźniki a tablice.	<b>2</b>
<b>Razem</b>		<b>10</b>

#### ZAJĘCIA LABORATORYJNE

<b>L1</b>	Zapoznanie się ze programistycznym środowiskiem pracy.	<b>2</b>
<b>L2</b>	Realizacja prostych algorytmów. Zastosowanie instrukcji sterujących i pętli.	<b>2</b>
<b>L3</b>	Tworzenie projektu i realizacja prostych algorytmów. Typy i reprezentacja danych.	<b>2</b>
<b>L4</b>	Tworzenie projektu i realizacja prostych algorytmów. Zmienne, operatory i wyrażenia.	<b>2</b>
<b>L5</b>	Formatowanie wejścia/wyjścia.	<b>2</b>
<b>L6</b>	Deklarowanie i definiowanie funkcji.	<b>2</b>
<b>L7</b>	Działanie na wskaźnikach i tablicach.	<b>2</b>
<b>L8</b>	Tworzenie struktur, unii i funkcji działających zmiennych struktur.	<b>3</b>
<b>L9</b>	Projekt własny	<b>3</b>
<b>Razem</b>		<b>20</b>

#### NARZĘDZIA DYDAKTYCZNE

<b>1</b>	Notebook z projektorem
<b>2</b>	Stanowiska komputerowe z oprogramowaniem dydaktycznym

#### SPOSOBY OCENY

##### FORMUJĄCA

<b>F2</b>	Odpowiedź ustna	EK1-EK6
<b>F3</b>	Wykonanie zadanie praktycznego	EK1-EK6

#### OBCIĄŻENIE PRACĄ STUDENTA

Forma aktywności	Średnia liczba godzin na zrealizowanie aktywności	
	semestr	razem
Godziny kontaktowe z nauczycielem	30	30
Samodzielne opracowanie zagadnień	20	20
Rozwiązywanie zadań domowych	20	20
Przygotowanie do wykładów i laboratoriów	20	20
<b>SUMA GODZIN W SEMESTRZE</b>	<b>90</b>	<b>90</b>
<b>PUNKTY ECTS W SEMESTRZE</b>	<b>3</b>	<b>3</b>

#### LITERATURA

##### PODSTAWOWA

<b>1</b>	B.W. Kernighan, D.M. Ritchie, Język ANSI C, WNT, Warszawa 2004.
<b>2</b>	S. Prata, Język C. Szkoła programowania, Wydawnictwo Robomatic, Warszawa 1999.
<b>3</b>	J. Liberty, Programowanie C#, Helion, Gliwice 2006.

4 N. Wirth, Wprowadzenie do programowania systematycznego. WNT, Warszawa 1978.

---

**PROWADZĄCY PRZEDMIOT**

1 dr hab. inż. Jerzy Garus, j.garus@amw.gdynia.pl

---

## Formy oceny

Efekt	Na ocenę 2	Na ocenę 3	Na ocenę 4	Na ocenę 5
EK1	<i>Student zna zasady tworzenia oprogramowania w języku C. Rozróżnia pojęcia kompilacja, interpretacja konsolidacja. Potrafi zdefiniować pojęcie algorytm. Zna metody opisu algorytmu. Zna zasady budowy schematu blokowego. Potrafi omówić podstawowe struktury programistyczne. Potrafi napisać i uruchomić prosty program w języku C.</i>			
	Student nie zna zasad tworzenia oprogramowania w języku C. Nie potrafi zdefiniować pojęcie algorytm. Nie zna metod opisu algorytmu. Zna zasady budowy schematu blokowego. Nie potrafi napisać i uruchomić prostego programu w języku C.	Student zna zasady tworzenia oprogramowania w języku C. Potrafi zdefiniować pojęcie algorytm. Zna zasady budowy schematu blokowego. Potrafi omówić podstawowe struktury programistyczne. Potrafi napisać i uruchomić prosty program w języku C.	Student zna zasady tworzenia oprogramowania w języku C. Rozróżnia pojęcia kompilacja, interpretacja konsolidacja. Potrafi zdefiniować pojęcie algorytm. Zna metody opisu algorytmu. Zna zasady budowy schematu blokowego. Potrafi omówić podstawowe struktury programistyczne. Potrafi napisać i uruchomić prosty program w języku C.	Student zna zasady tworzenia oprogramowania w języku C. Doskonale rozróżnia pojęcia kompilacja, interpretacja konsolidacja. Potrafi zdefiniować pojęcie algorytm. Zna metody opisu algorytmu. Zna doskonale zasady budowy schematu blokowego. Potrafi omówić podstawowe struktury programistyczne. Potrafi biegle napisać i uruchomić prosty program w języku C.
EK2	<i>Student potrafi omówić instrukcje złożone w instrukcjach sterujących. Potrafi wyjaśnić działanie instrukcji warunkowej if ... else, pętli while i for, wyboru (switch) oraz zasad przerywania i kontynuacji wykonywania pętli. Zna operatory matematyczne, relacyjne, logiczne, bitowe i operator przypisania. Zna priorytety operatorów.</i>			
	Student nie potrafi wyjaśnić działanie instrukcji warunkowej if ... else, pętli while i for, wyboru (switch). Słabo zna operatory matematyczne, relacyjne, logiczne, bitowe i operator przypisania.	Student potrafi omówić instrukcje złożone w instrukcjach sterujących. Potrafi wyjaśnić działanie instrukcji warunkowej if ... else, pętli while i for, wyboru (switch). Zna operatory matematyczne, relacyjne, logiczne, bitowe i operator przypisania.	Student potrafi omówić instrukcje złożone w instrukcjach sterujących. Potrafi wyjaśnić działanie instrukcji warunkowej if ... else, pętli while i for, wyboru (switch) oraz zasad przerywania i kontynuacji wykonywania pętli. Zna operatory matematyczne, relacyjne, logiczne, bitowe i operator przypisania. Zna priorytety operatorów.	Student potrafi omówić instrukcje złożone w instrukcjach sterujących. Doskonale potrafi wyjaśnić działanie instrukcji warunkowej if ... else, pętli while i for, wyboru (switch) oraz zasad przerywania i kontynuacji wykonywania pętli. Zna i potrafi zastosować praktycznie operatory matematyczne, relacyjne, logiczne, bitowe i operator przypisania. Zna priorytety operatorów.
EK3	<i>Student potrafi wymienić złożone typy danych. Zna zasady tworzenia tablic liczbowych i znakowych. Potrafi inicjalizować tablice. Zna zasady konwersji łańcuchów do liczb. Zna przeznaczenie i sposób deklaracji struktury.</i>			
	Student nie zna złożonych typów danych. Nie zna zasad tworzenia tablic liczbowych i znakowych. Nie zna zasady konwersji łańcuchów do liczb. Zna typy struktur.	Student potrafi wymienić złożone typy danych. Słabo zna zasady tworzenia tablic liczbowych i znakowych. Słabo zna zasady konwersji łańcuchów do liczb. Zna przeznaczenie struktury.	Student potrafi wymienić złożone typy danych. Zna zasady tworzenia tablic liczbowych i znakowych. Potrafi inicjalizować tablice. Zna zasady konwersji łańcuchów do liczb. Zna sposób deklaracji struktury.	Student potrafi wymienić złożone typy danych. Doskonale zna zasady tworzenia tablic liczbowych i znakowych. Potrafi inicjalizować tablice. Zna zasady konwersji łańcuchów do liczb. Zna przeznaczenie i sposób deklaracji struktury.

<b>EK4</b>	<i>Student zna zasady wprowadzania i wyprowadzania danych. Potrafi sterować formatem. Zna i potrafi zastosować praktycznie podstawowe funkcje z biblioteki STDIO. Zna zasady obsługi plików oraz plikowe funkcje wejścia/wyjścia.</i>			
	Student zna zasady wprowadzania i wyprowadzania danych. Nie potrafi sterować formatem. Zna, lecz nie potrafi zastosować praktycznie podstawowych funkcji z biblioteki STDIO. Nie zna zasad obsługi plików.	Student słabo zna zasady wprowadzania i wyprowadzania danych. Potrafi zastosować praktycznie podstawowe funkcje z biblioteki STDIO. Zna podstawowe zasady obsługi plików.	Student zna zasady wprowadzania i wyprowadzania danych. Potrafi sterować formatem. Zna i potrafi zastosować praktycznie podstawowe funkcje z biblioteki STDIO. Zna zasady obsługi plików oraz plikowe funkcje wejścia/wyjścia.	Student doskonale zna zasady wprowadzania i wyprowadzania danych. Potrafi sterować formatem. Bardzo dobrze zna i potrafi zastosować praktycznie podstawowe funkcje z biblioteki STDIO. Zna doskonale zasady obsługi plików oraz plikowe funkcje wejścia/wyjścia.
<b>EK5</b>	<i>Student rozumie pojęcie funkcja. Zna zasady przekazywania parametrów do funkcji i zwracania wyniku funkcji. Wie co to prototyp funkcji (typ funkcyjny) oraz czas życia i zakres ważności nazwy. Potrafi praktycznie korzystać z funkcji bibliotecznych.</i>			
	Student rozumie pojęcie funkcja. Nie zna zasady przekazywania parametrów do funkcji i zwracania wyniku funkcji. Wie co to prototyp funkcji. Nie potrafi praktycznie korzystać z funkcji bibliotecznych.	Student rozumie pojęcie funkcja. Zna zasady przekazywania parametrów do funkcji i zwracania wyniku funkcji. Rozumie pojęcie prototyp funkcji. Potrafi praktycznie korzystać z funkcji bibliotecznych.	Student rozumie pojęcie funkcja. Zna zasady przekazywania parametrów do funkcji i zwracania wyniku funkcji. Wie co to prototyp funkcji oraz czas życia i zakres ważności nazwy. Potrafi praktycznie korzystać z funkcji bibliotecznych.	Student rozumie pojęcie funkcja. Doskonale zna zasady przekazywania parametrów do funkcji i zwracania wyniku funkcji. Wie co to prototyp funkcji oraz czas życia i zakres ważności nazwy. Bardzo dobrze potrafi praktycznie korzystać z funkcji bibliotecznych.
<b>EK6</b>	<i>Student zna przeznaczenie wskaźników. Zna działania na wskaźnikach. Rozumie pojęcie dynamiczna alokacja pamięć. Potrafi zastosować wskaźniki do przekazywania parametrów oraz dostępu do tablic i składników struktury. Potrafi napisać i uruchomić złożony program w języku C.</i>			
	Student zna działania na wskaźnikach. Nie rozumie pojęcia dynamiczna alokacja pamięć. Nie potrafi zastosować wskaźniki do przekazywania parametrów oraz dostępu do tablic i składników struktury. Z trudnością pisze i uruchamia program w języku C.	Student zna przeznaczenie wskaźników. Zna działania na wskaźnikach. Słabo potrafi zastosować wskaźniki do przekazywania parametrów oraz dostępu do tablic i składników struktury. Potrafi napisać i uruchomić niezbyt złożony program w języku C.	Student zna przeznaczenie wskaźników. Zna działania na wskaźnikach. Rozumie pojęcie dynamiczna alokacja pamięć. Potrafi zastosować wskaźniki do przekazywania parametrów oraz dostępu do tablic i składników struktury. Potrafi napisać i uruchomić złożony program w języku C.	Student zna przeznaczenie wskaźników. Doskonale zna działania na wskaźnikach. Rozumie pojęcie dynamiczna alokacja pamięć. Biegle potrafi zastosować wskaźniki do przekazywania parametrów oraz dostępu do tablic i składników struktury. Biegle potrafi napisać i uruchomić złożony program w języku C.