

I. KARTA PRZEDMIOTU

1. Nazwa przedmiotu: **PODSTAWY PROGRAMOWANIA**
2. Kod przedmiotu: **Ovi1**
3. Jednostka prowadząca: **Wydział Mechaniczno-Elektryczny**
4. Kierunek: **Automatyka i Robotyka**
5. Specjalność: **Elektroautomatyka Okrętowa**
6. Moduł: **treści podstawowych**
7. Poziom studiów: **I stopnia**
8. Forma studiów: **niestacjonarne**
9. Semestr studiów: **II**
10. Profil: **ogólnoakademicki**
11. Prowadzący: **dr hab. inż. Jerzy Garus**

CEL PRZEDMIOTU

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

WYMAGANIA WSTĘPNE W ZAKRESIE WIEDZY, UMIEJĘTNOŚCI I INNYCH KOMPETENCJI

1	Wiedza z zakresu matematyki i technologii informacyjnej.
2	Znajomość pozycyjnych systemów zapisu liczb.

EFEKTY KSZTAŁCENIA

EK1	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.
EK2	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.
EK3	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.
EK4	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.
EK5	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.
EK6	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
W1	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.	1

W2	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.	1
W3	Typy danych: całkowite, znakowe i zmiennoprzecinkowe. Konwersja typów. Złożone typy danych: tablice liczbowe i znakowe, rekordy i unie.	1
W4	Operacje wejścia/wyjścia: operacje wejścia/wyjścia (biblioteka stdio); sterowanie formatem, operacje wejścia/wyjścia na plikach, obsługa błędów.	1
W5	Pojęcie funkcji, zwracanie wyniku, przekazywanie parametrów, prototyp funkcji, zmienne lokalne i globalne, funkcje biblioteczne.	1
W6	Zarządzanie pamięcią: wskaźniki, dynamiczna alokacja pamięci, operator sizeof(), wskaźniki a tablice.	1
Razem		6

ZAJĘCIA LABORATORYJNE

L1	Zapoznanie się ze programistycznym środowiskiem pracy.	1
L2	Realizacja prostych algorytmów. Zastosowanie instrukcji sterujących i pętli.	1
L3	Tworzenie projektu i realizacja prostych algorytmów. Typy i reprezentacja danych.	1
L4	Tworzenie projektu i realizacja prostych algorytmów. Zmienne, operatory i wyrażenia.	1
L5	Formatowanie wejścia/wyjścia.	1
L6	Deklarowanie i definiowanie funkcji.	1
L7	Działanie na wskaźnikach i tablicach.	1
L8	Tworzenie struktur, unii i funkcji działających zmiennych struktur.	3
L9	Projekt własny	2
Razem		12

NARZĘDZIA DYDAKTYCZNE

1	Notebook z projektorem
2	Stanowiska komputerowe z oprogramowaniem dydaktycznym

SPOSOBY OCENY

FORMUJĄCA

F2	Odpowiedź ustna	EK1-EK6
F3	Wykonanie zadanie praktycznego	EK1-EK6

OBCIĄŻENIE PRACĄ STUDENTA

Forma aktywności	Średnia liczba godzin na zrealizowanie aktywności	
	semestr	razem
udział w wykładach	6	6
udział w zajęciach laboratoryjnych	12	12
Godziny kontaktowe z nauczycielem	30	30
Samodzielne opracowanie zagadnień	10	10
Rozwiązywanie zadań domowych	10	10
Przygotowanie do wykładów i laboratoriów	20	20
SUMA GODZIN W SEMESTRZE	88	88
PUNKTY ECTS W SEMESTRZE	3	3

LITERATURA

PODSTAWOWA

1	B.W. Kernighan, D.M. Ritchie, Język ANSI C, WNT, Warszawa 2004.
----------	---

2 S. Prata, Język C. Szkoła programowania, Wydawnictwo Robomatic, Warszawa 1999.

3 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.

EK4	<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.
EK5	<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.
EK6	<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.