



## IV. Opis programu studiów

### 3. KARTA PRZEDMIOTU

Kod przedmiotu	<b>M#1-N1-IP-107</b>
Nazwa przedmiotu	<b>Programowanie w języku C</b>
Nazwa przedmiotu w języku angielskim	<b>Programming in C language</b>
Obowiązuje od roku akademickiego	<b>2020/2021</b>

### USYTUOWANIE MODUŁU W SYSTEMIE STUDIÓW

Kierunek studiów	<b>INFORMATYKA PRZEMYSŁOWA</b>
Poziom kształcenia	<b>I stopień</b>
Profil studiów	<b>ogólnoakademicki</b>
Forma i tryb prowadzenia studiów	<b>studia niestacjonarne</b>
Zakres	<b>komputerowe systemy przemysłowe</b>
Jednostka prowadząca przedmiot	<b>Katedra Technologii Mechanicznej i Metrologii</b>
Koordynator przedmiotu	<b>mgr inż. A. Arciszewski</b>
Zatwierdził	

### OGÓLNA CHARAKTERYSTYKA PRZEDMIOTU

Przynależność do grupy/bloku przedmiotów	<b>przedmiot kierunkowy</b>
Status przedmiotu	<b>obowiązkowy</b>
Język prowadzenia zajęć	polski
Usytuowanie modułu w planie studiów - semestr	<b>semestr 1</b>
Wymagania wstępne	
Egzamin (TAK/NIE)	NIE
Liczba punktów ECTS	<b>3</b>

Forma prowadzenia zajęć	wykład	ćwiczenia	laboratorium	projekt	seminarium
Liczba godzin w semestrze	<b>9</b>		<b>18</b>		

**EFEKTY UCZENIA SIĘ**

Kategoria	Symbol efektu	Efekty kształcenia	Odniesienie do efektów kierunkowych
Wiedza	W01	Student zna podstawowe struktury danych, konstrukcje programistyczne oraz metody weryfikacji poprawności programów.	IP1_W11 IP1_W25
	W02	Student zna instrukcje sterujące przepływem danych w programie: if, if-else, switch, instrukcje pętlowe while, do-while, for, pętle zagnieżdżone, instrukcje przerwań break i continue.	IP1_W11 IP1_W25
	W03	Student zna funkcje, zmienne lokalne i globalne.	IP1_W11 IP1_W25
	W04	Student zna łańcuchy, tablice jedno i wielowymiarowe, wskaźniki.	IP1_W11 IP1_W25
	W05	Student zna struktury, operacje na strukturach.	IP1_W11 IP1_W25
	W06	Student zna dynamiczny przydział pamięci.	IP1_W11 IP1_W25
Umiejętności	U01	Student potrafi czytać ze zrozumieniem i konstruować algorytmy rozwiązujące wybrane problemy matematyczne i zapisywać je w postaci kodu źródłowego języka C, potrafi korzystać ze standardowych funkcji bibliotecznych tego języka.	IP1_U01 IP1_U02 IP1_U25
	U02	Student potrafi korzystać z instrukcji sterujących przepływem danych w programie: if, if-else, switch. Student potrafi korzystać z pętli while, do-while, for oraz instrukcji przerwań break i continue.	IP1_U25
	U03	Student potrafi korzystać z funkcji, tablic i łańcuchów znaków.	IP1_U25
	U04	Student potrafi korzystać z zaawansowanych technik programistycznych takich jak: dynamiczne zarządzanie pamięcią, dynamiczne struktury danych, rekurencja.	IP1_U25
Kompetencje społeczne	K01	Rozumie potrzebę i zna możliwości ciągłego doskonalenia (studia II i III stopnia, studia podyplomowe, kursy) mającego na celu podnoszenie kompetencji zawodowych, osobistych i społecznych.	IP1_K01
	K02	Ma świadomość odpowiedzialności za pracę własną oraz gotowość podporządkowania się zasadom pracy w zespole i ponoszenia odpowiedzialności za wspólnie realizowane zadania.	IP1_K04

**TREŚCI PROGRAMOWE**

Forma zajęć*	Treści programowe
wykład	1. Definicje pojęć: program, translator, kompilator, interpreter, moduł programowy. Programowanie algorytmiczne, strukturalne, obiektowe. Podział języków programowania ze względu na paradygmat programowania. Język C – wybrany język programowania.
	2. Struktura programu w języku C. Typy danych, operatory i wyrażenia. Operacje wejścia i wyjścia.
	3. Instrukcje sterujące przepływem danych w programie: if, if-else, switch. Zagnieżdżanie.
	4. Instrukcje pętlowe while, do-while, for. Pętle zagnieżdżone. Instrukcje przerwań break i continue.

	5.Funkcje. Zmienne lokalne i globalne. Deklaracja funkcji. Zwracanie rezultatu przez funkcję. Przekazywanie zmiennych do funkcji.
	6.Tablice jedno i wielowymiarowe. Wskaźniki. Wskaźniki do tablic, wskaźniki do funkcji, wskaźniki do wskaźników. Tablice wskaźników. Arytmetyka wskaźników.
	7.Łańcuchy, wyczytywanie, wyświetlanie, funkcje łańcuchowe.
	8.Pliki. Deklarowanie, otwieranie i zamykanie. Dodawanie danych do pliku.
	9.Struktury: Definiowanie struktur. Operacje na strukturach.
	10.Dynamiczny przydział pamięci funkcje malloc(), calloc().
laboratorium	1.Środowisko programistyczne CodeBlocks – 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.Pliki. Deklarowanie, otwieranie i zamykanie. Dodawanie danych do pliku.
	7.Struktury: Definiowanie struktur. Operacje na strukturach.
	8.Dynamiczny przydział pamięci funkcje malloc(), calloc().

### **METODY WERYFIKACJI EFEKTÓW UCZENIA SIĘ**

Symbol efektu	Metody sprawdzania efektów kształcenia <i>(zaznaczyć X)</i>					
	Egzamin ustny	Egzamin pisemny	Kolokwium	Projekt	Sprawozdanie	Inne
W01			x			
W02			x			
W03			x			
W04			x			
W05			x			
W06			x			
U01			x			
U02			x			
U03			x			
U04			x			
K01						x
K02						x

### **FORMA I WARUNKI ZALICZENIA**

Forma zajęć*	Forma zaliczenia	Warunki zaliczenia
wykład	<b>zaliczenie z oceną</b>	<i>Uzyskanie co najmniej 50% punktów z kolokwium</i>
laboratorium	zaliczenie z oceną	<i>Uzyskanie co najmniej 50% punktów z kolokwiów w trakcie zajęć</i>

### **NAKŁAD PRACY STUDENTA**

Bilans punktów ECTS							
Lp.	Rodzaj aktywności	Obciążenie studenta					Jednostka
		W	C	L	P	S	
1.	Udział w zajęciach zgodnie z planem studiów	9		18			h
2.	Inne (konsultacje, egzamin)	2		2			h
3.	<b>Razem przy bezpośrednim udziale nauczyciela akademickiego</b>	<b>31</b>					h
4.	<b>Liczba punktów ECTS, którą student uzyskuje przy bezpośrednim udziale nauczyciela akademickiego</b>	<b>1,2</b>					ECTS
5.	<b>Liczba godzin samodzielnej pracy studenta</b>	<b>44</b>					h
6.	<b>Liczba punktów ECTS, którą student uzyskuje w ramach samodzielnej pracy</b>	<b>1,8</b>					ECTS
7.	<b>Nakład pracy związany z zajęciami o charakterze praktycznym</b>	<b>50</b>					h
8.	<b>Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym</b>	<b>2,0</b>					ECTS
9.	<b>Sumaryczne obciążenie pracą studenta</b>	<b>75</b>					h
10.	<b>Punkty ECTS za moduł</b> <i>1 punkt ECTS=25 godzin obciążenia studenta</i>	<b>3</b>					ECTS

## LITERATURA

1. Prata Stephen.: Język C. Szkoła programowania. Helion