



KARTA MODUŁU / KARTA PRZEDMIOTU

Kod modułu	
Nazwa modułu	Programowanie komputerów C (1)
Nazwa modułu w języku angielskim	Programming In C 1
Obowiązuje od roku akademickiego	2012/2013

A. USYTUOWANIE MODUŁU W SYSTEMIE STUDIÓW

Kierunek studiów	Elektrotechnika
Poziom kształcenia	I stopień (I stopień / II stopień)
Profil studiów	Ogólno akademicki (ogólno akademicki / praktyczny)
Forma i tryb prowadzenia studiów	stacjonarne (stacjonarne / niestacjonarne)
Specjalność	Komputerowe Systemy Pomiarowe
Jednostka prowadząca moduł	Katedra Elektrotechniki i Systemów Pomiarowych
Koordynator modułu	dr inż. Cezary Siwoń
Zatwierdził:	

B. OGÓLNA CHARAKTERYSTYKA PRZEDMIOTU

Przynależność do grupy/bloku przedmiotów	kierunkowy (podstawowy / kierunkowy / inny HES)
Status modułu	obowiązkowy (obowiązkowy / nieobowiązkowy)
Język prowadzenia zajęć	Polski
Usytuowanie modułu w planie studiów - semestr	IV
Usytuowanie realizacji przedmiotu w roku akademickim	Letni (semestr zimowy / letni)
Wymagania wstępne	Brak wymagań (kody modułów / nazwy modułów)
Egzamin	nie (tak / nie)
Liczba punktów ECTS	3

Forma prowadzenia zajęć	wykład	ćwiczenia	laboratorium	projekt	inne
w semestrze	30	15			



C. EFEKTY KSZTAŁCENIA I METODY SPRAWDZANIA EFEKTÓW KSZTAŁCENIA

Cel modułu	Zapoznanie studentów z podstawami tworzenia oprogramowania w języku C, z narzędziami do tworzenia oprogramowania, podstawowymi technikami tworzenia oprogramowania i algorytmami. <i>(3-4 linijki)</i>
-------------------	---

Symbol efektu	Efekty kształcenia	Forma prowadzenia zajęć (w/ć/l/p/inne)	odniesienie do efektów kierunkowych	odniesienie do efektów obszarowych
W_01	Ma podstawową wiedzę o składni języka C, słowach kluczowych, operatorach, instrukcjach, funkcjach, bibliotekach	Wykład	K_W03 K_W15	T1A_W02
W_02	Zna podstawy tworzenia oprogramowania w języku C	Wykład	K_W15 K_W03	T1A_W02 T1A_W07
U_01	Umie analizować i rozumie proste programy w języku C	Ćwiczenia	K_U17 K_015	T1A_U13
U_02	Umie tworzyć elementarne programy w języku C	Ćwiczenia	K_U17	T1A_U07
U_03	Potrafi korzystać z narzędzi do uruchamiania, debugowania i tworzenia programów	Ćwiczenia	K_U17 K_U13	T1A_U15
K_01	Potrafi współdziałać i pracować w grupie	Ćwiczenia	K_K04	T1A_K03

Treści kształcenia:

1. Treści kształcenia w zakresie wykładu

Nr wykładu	Treści kształcenia	Odniesienie do efektów kształcenia dla modułu
1	Historia. Charakterystyka języka. Podstawowe elementy języka – zbiór znaków, słowa kluczowe, identyfikatory, komentarze. Struktura programu w przykładach. Zmienne i ich typy – typy podstawowe: całkowite, rzeczywiste. Typy podstawowe – typ wyliczeniowy, tablice, łańcuchy. Typy złożone – struktura, unia. Wskaźniki.	W_01
2, 3	Wyrażenia – budowa, priorytety i wiązanie, rodzaje operatorów i wartości. Operatory arytmetyczne, znaku, bitowe, przesunięcia i logiczne. Operatory relacji, porównania, przypisania, Operator warunkowy, wyliczenia, pobrania adresu, odwołania do wartości wskazywanej. Pobrania rozmiaru, konwersji, dostępu do składowych, indeksowania, wywołania funkcji.	W_01
4, 5	Instrukcje w języku C – z etykietą, pusta, złożona, wyrażeniowa, warunkowa, wyboru. Instrukcje iteracyjne: for, while i do while. Instrukcje skoku – got, break, continue, return. Funkcje – deklaracja (prototyp) i definicja. Przekazywanie argumentów różnego typu do funkcji. Tablice a wskaźniki. Postać funkcji main – przekazywanie parametrów z linii poleceń. Funkcje o nieokreślonej liczbie parametrów.	W_01
6, 7	Konwersje typów: jawne i niejawne. Zmienne: rodzaje, zasięg deklaracji, widoczność i czas istnienia. Zmienne dynamiczne: tworzenie, eksploatacja i usuwanie w języku C.	W_01
8, 9	Funkcje wejścia-wyjścia sformatowanego i niesformatowanego.	W_01
10, 11	Wybrane funkcje z biblioteki standardowej: operacje na łańcuchach, funkcje matematyczne i inne. Przegląd ważniejszych problemów pojawiających się w trakcie tworzenia kodu źródłowego. Preprocesor: rola i działanie. Dyrektywy i makroinstrukcje. Problemy stosowania makroinstrukcji. Proces tworzenia pliku	W_01



	wykonywalnego.	
12, 13	Tworzenie programów wielomodułowych. Podstawowe zasady bezpiecznego i dobrego tworzenia oprogramowania. Konwencje nazewnicze. Przegląd kompilatorów, środowisk programistycznych i dodatkowych narzędzi do tworzenia programów w C.	W_01 W_02
14, 15	Implementacja podstawowych algorytmów w języku C: sortowanie, wyszukiwanie. Implementacja struktur danych takich jak stos, kolejka, lista	W_01 W_02 U_01

2. Treści kształcenia w zakresie ćwiczeń

Nr zajęć ćwicz.	Treści kształcenia	Odniesienie do efektów kształcenia dla modułu
1, 2	Środowisko programistyczne DevC++, funkcja main, tworzenie programu z użyciem wyrażeń i instrukcji warunkowej	U_02 U_03 K_01
3, 4	Instrukcje iteracyjne i tablice	U_01 U_02 K_01
5, 6	Definiowanie funkcji i korzystanie z funkcji bibliotecznych	U_01 U_02 K_01
7, 8	Operacje wejścia-wyjścia	U_01 U_02 K_01

3. Treści kształcenia w zakresie zadań laboratoryjnych

Nr zajęć lab.	Treści kształcenia	Odniesienie do efektów kształcenia dla modułu

4. Charakterystyka zadań projektowych

5. Charakterystyka zadań w ramach innych typów zajęć dydaktycznych

Metody sprawdzania efektów kształcenia

Symbol efektu	Metody sprawdzania efektów kształcenia (sposób sprawdzenia, w tym dla umiejętności – odwołanie do konkretnych zadań projektowych, laboratoryjnych, itp.)
W_01	Kolokwium
W_02	Kolokwium
U_01	Kolokwium
U_02	Sprawozdanie z ćwiczeń 1, 2, 3, 4
U_03	Kolokwium
K_01	Sprawozdanie z ćwiczeń 1, 2, 3, 4



D. NAKŁAD PRACY STUDENTA

Bilans punktów ECTS		
	Rodzaj aktywności	obciążenie studenta
1	Udział w wykładach	30
2	Udział w ćwiczeniach	15
3	Udział w laboratoriach	
4	Udział w konsultacjach (2-3 razy w semestrze)	2
5	Udział w zajęciach projektowych	
6	Konsultacje projektowe	
7	Udział w egzaminie	
8		
9	Liczba godzin realizowanych przy bezpośrednim udziale nauczyciela akademickiego	47 (suma)
10	Liczba punktów ECTS, którą student uzyskuje na zajęciach wymagających bezpośredniego udziału nauczyciela akademickiego (1 punkt ECTS=25-30 godzin obciążenia studenta)	1,88
11	Samodzielne studiowanie tematyki wykładów	10
12	Samodzielne przygotowanie się do ćwiczeń	10
13	Samodzielne przygotowanie się do kolokwium	4
14	Samodzielne przygotowanie się do laboratoriów	
15	Wykonanie sprawozdań	4
15	Przygotowanie do kolokwium końcowego z laboratorium	
17	Wykonanie projektu lub dokumentacji	
18	Przygotowanie do egzaminu	
19		
20	Liczba godzin samodzielnej pracy studenta	28 (suma)
21	Liczba punktów ECTS, którą student uzyskuje w ramach samodzielnej pracy (1 punkt ECTS=25-30 godzin obciążenia studenta)	1,12
22	Sumaryczne obciążenie pracą studenta	75
23	Punkty ECTS za moduł 1 punkt ECTS=25-30 godzin obciążenia studenta	3
24	Nakład pracy związany z zajęciami o charakterze praktycznym Suma godzin związanych z zajęciami praktycznymi	17
25	Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym 1 punkt ECTS=25-30 godzin obciążenia studenta	0,68

E. LITERATURA

Wykaz literatury	1. Kernighan B., Ritchie D.: Język ANSI C, programowanie, Helion 2010 2. K. N. King , Język C. Nowoczesne programowanie. Wydanie II, luty 2011, Helion 3. Zalewski A.: Programowanie w językach C i C++ z wykorzystaniem pakietu Borland C++, seria BUM, Wyd. NAKOM, Poznań 1994.
Witryna WWW modułu/przedmiotu	