

KARTA PRZEDMIOTU

1. Informacje ogólne

Nazwa przedmiotu i kod (wg planu studiów):	Programowanie niskopoziomowe, C2
Nazwa przedmiotu (j. ang.):	Low Level Programming
Kierunek studiów:	Informatyka
Specjalność/specjalizacja:	
Poziom kształcenia:	studia I stopnia
Profil kształcenia:	praktyczny (P)
Forma studiów:	studia stacjonarne
Obszar kształcenia:	nauki techniczne (wg wykazu)
Dziedzina:	nauki techniczne (wg wykazu)
Dyscyplina nauki:	(wg wykazu)
Koordinator przedmiotu:	dr Marcin Skuba

2. Ogólna charakterystyka przedmiotu

Przynależność do modułu:	kierunkowego
Status przedmiotu:	obowiązkowy
Język wykładowy:	polski
Rok studiów, semestr:	I, 1
Forma i wymiar zajęć według planu studiów:	stacjonarne - wykład 15 h, ćw. laboratoryjne 30 h
Interesariusze i instytucje partnerskie (nieobowiązkowe)	
Wymagania wstępne / Przedmioty wprowadzające:	Dobra umiejętność posługiwania się komputerem brak

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):	2 (A + B)	stacjonarne
A. Liczba godzin wymagających bezpośredniego udziału nauczyciela (kontaktowych, w czasie rzeczywistym, w tym testy, egzaminy etc) z podziałem na typy zajęć oraz całkowita liczba punktów ECTS osiągniętych na tych zajęciach	obecność na wykładach obecność na ćwiczeniach laboratoryjnych udział w konsultacjach W sumie: ECTS	15 15 2 32 1
B. Poszczególne typy zadań do samokształcenia studenta (niewymagających bezpośredniego udziału nauczyciela) wraz z planowaną średnią liczbą godzin na każde i sumaryczną liczbą ECTS (np. praca w bibliotece, w sieci, na platformie e-learningowej, w laboratorium, praca nad projektem końcowym, przygotowanie ogólne; suma poszczególnych godzin powinna zgadzać się z liczbą ogólną)	przygotowanie ogólne przygotowanie do ćwiczeń laboratoryjnych wykonanie sprawozdań przygotowanie do kolokwium zaliczeniowego praca w bibliotece praca w sieci w sumie: ECTS	2 10 5 5 4 4 30 1
C. Liczba godzin praktycznych/laboratoryjnych w ramach przedmiotu 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):	udział w ćwiczeniach laboratoryjnych praca praktyczna samodzielna w sumie: ECTS	15 30 45 1,5

4. Opis przedmiotu

Cel przedmiotu: Celem przedmiotu jest nabycie praktycznych umiejętności zarządzania pracą procesora w niskopoziomym języku assembler.
Metody dydaktyczne: wykład - pokaz, laboratorium - zadania problemowe
Treści kształcenia (w rozbiciu na formę zajęć (jeśli są różne formy) i najlepiej w punktach): Wykłady: <ol style="list-style-type: none"> 1. Wprowadzenie do środowiska niskopoziomowego, kompilatory, typy programów, architektura komputera, systemy liczbowe, 2. Programy typu COM, Rejestry procesora, debugowanie kodu programu, <ol style="list-style-type: none"> 1. Mikroprocesor 8086 i kontroler 8288. 3. Przerwania, 4. Rozkazy arytmetyczno logiczne, 5. Stos, przesunięcia bitowe, 6. Rozkazy sterujące bezwarunkowe, etykiety i procedury, 7. Instrukcje warunkowe, 8. Pętle, 9. Tablice, 10. Argumenty funkcji, 11. Dostęp do plików na dysku, 12. Struktura programu typu EXE.

13. Protokół ModBus.
14. Analiza przykładowych programów napisanych w języku assembler.
15. Asemlacja warunkowa, podsumowanie.

Ćwiczenia laboratoryjne:

1. Instalacja narzędzi, kompilacja programu typu COM.
2. Rejestru procesora,
3. Przerwania,
4. Stos,
5. Rozkazy arytmetyczno-logiczne,
6. Rozkazy bezwarunkowe, definiowanie procedur,
7. Rozkazy warunkowe,
8. Pętle,
9. Tablice,
10. Argumenty funkcji,
11. Asemlacja warunkowa,
12. Operacje na plikach
13. Programy typu EXE

5. Efekty kształcenia i sposoby weryfikacji

Efekty kształcenia				
Efekt przedmiotu	Student, który zaliczył przedmiot (spełnił minimum wymagań)			Efekt kierunkowy
C2_K_W01 C2_K_W02	Wiedza: Student zna zasadę wykonywania instrukcji kodu źródłowego języka assembler. Student zna architekturę oraz listę rozkazów procesora 8086.			K_W06 K_W16
C2_K_U01 C2_K_U02 C2_K_U03	Umiejętności Student potrafi tworzyć kod źródłowy oraz kompilować, konsolidować i debugować program w języku assembler, Student umie wykorzystać podstawowe instrukcje języka assembler. Student potrafi wywoływać przerwania wg opisu dokumentacji oraz umie wykorzystać stos.			K_U01 K_U17 K_U13 K_U03
C2_K_W01	Kompetencje społeczne Student rozumie potrzebę wykorzystania poznanej na zajęciach wiedzy i umiejętności do programowania układów mikroprocesorowych.			K_K01
Sposoby weryfikacji efektów kształcenia:				
Lp.	Efekt przedmiotu	Sposób weryfikacji	Ocena formująca poniżej	Ocena końcowa
1	C2_K_W01	Zadania problemowe na zajęciach laboratoryjnych, kolokwium	Praktyczny sprawdzian wiadomości, ocena z sprawozdania	Egzamin
2	C2_K_W02			
3	C2_K_U01	Zadania problemowe na zajęciach laboratoryjnych, kolokwium	Praktyczny sprawdzian umiejętności, ocena z sprawozdania	
4	C2_K_U02			

5	C2_K_U03			
6	C2_K_K01	Wykład	Sprawdzian ustny	
Kryteria oceny (oceny 3,0 powinny być równoważne z efektami kształcenia, choć mogą być bardziej szczegółowo opisane):				
w zakresie wiedzy				Efekt kształcenia
Na ocenę 3,0	Student zna zasadę wykonywania instrukcji kodu źródłowego języka assembler.		C2_K_W01	
Na ocenę 5,0	Student zna zasadę wykonywania instrukcji kodu źródłowego języka assembler, rozumie zasadę działania koprocatora.			
Na ocenę 3,0	Student zna architekturę oraz listę rozkazów procesora 8086.		C2_K_W02	
Na ocenę 5,0	Student zna architekturę oraz listę rozkazów procesora 8086 oraz rozumie zasady działania elementarnych układów cyfrowych mikroprocesora.			
w zakresie umiejętności				
Na ocenę 3,0	Student potrafi tworzyć kod źródłowy oraz kompilować, konsolidować i debugować program w języku assembler (plik typu COM).		C2_K_U01	
Na ocenę 5,0	Student potrafi tworzyć kod źródłowy oraz kompilować, konsolidować i debugować program w języku assembler (pliki typu COM i EXE). Zna struktury pliku COM i EXE.			
Na ocenę 3,0	Student umie wykorzystać podstawowe instrukcje języka assembler. Potrafi używać instrukcji pętli oraz instrukcji skoku warunkowego.		C2_K_U02	
Na ocenę 5,0	Student zna instrukcje języka assembler. Potrafi używać instrukcji pętli, skoku warunkowego, tworzyć procedury.			
Na ocenę 3,0	Student potrafi wywoływać proste przerwania wg opisu dokumentacji oraz umie wykorzystać stos.		C2_K_U03	
Na ocenę 5,0	Student potrafi wywoływać rozbudowane przerwania (dysku, karty graficznej) wg opisu dokumentacji oraz umie wykorzystać stos do przekazywania argumentów do procedury.			
w zakresie kompetencji społecznych				
Na ocenę 3,0	Student rozumie potrzebę wykorzystania poznanej na zajęciach wiedzy i umiejętności do programowania prostych układów mikroprocesorowych.		C2_K_K01	
Na ocenę 5,0	Student rozumie potrzebę wykorzystania poznanej na zajęciach wiedzy i umiejętności do programowania zaawansowanych układów mikroprocesorowych.			
Kryteria oceny końcowej				
kolokwia: 50 % samodzielne wykonanie ćwiczeń laboratoryjnych: 20%, aktywność za zajęciach: 15%, ocena ze sprawozdania: 15%,				
Zalecana literatura				
Literatura podstawowa:		1. Kruk S., Programowanie w języku Assembler , W-wa, Mikom, PLJ. 1993. 2. Wajs W. Organizacja I Architektura Komputera PC. Wyd. AGH Kraków 2010.		

Literatura uzupełniająca:	<ol style="list-style-type: none"> 1. J. Augustyn, Asemblery, Kraków, Wydawnictwo IGSMiE PAN , 2006 2. m. Gawrylczyk, Efekty graficzne w asemblerze, Gliwice : Helion , 1996 3. J. Dauntemann Zrozumieć asembler, Warszawa : "Translator" , 1993 4. A. Rydzewski, Mikrokomputery jednoukładowe rodziny MCS-51. 5. G. Syck, Turbo Assembler - Biblia użytkownika, LTP Oficyna Wydawnicza, 2002
----------------------------------	--

Informacje dodatkowe:

Dodatkowe obowiązki prowadzącego wraz z szacowaną całkowitą liczbą godzin:
Konsultacje – 20 godzin
Przygotowanie wykładów, zajęć laboratoryjnych - 15
Przygotowanie i poprawa egzaminu – 5 godzin
W sumie: 40 godzin