



Akademia Nauk Stosowanych
im. Hipolita Cegielskiego w Gnieźnie Uczelnia Państwowa

SYLABUS

Pozycja przedmiotu w planie:		R.I/S.II - 8
1. OGÓLNY OPIS PRZEDMIOTU		
1	Nazwa modułu	Moduł treści kierunkowych
2	Nazwa przedmiotu	Architektura komputerów
3	Kierunek studiów	Informatyka
4	Poziom studiów	pierwszy
5	Forma studiów	stacjonarne
6	Profil studiów	praktyczny
7	Rok studiów	pierwszy
8	Semestr przedmiotu	drugi
9	Jednostka prowadząca kierunek studiów	Instytut Nauk Technicznych
10	Liczba punktów ECTS	3
11	Sposób zaliczenia:	wykład: egzamin laboratorium: zaliczenie z oceną
12	Imię i nazwisko nauczyciela (li) akademickiego (ich), stopień lub tytuł naukowy, adres e-mail	mgr inż. Mateusz Leszek <i>m.leszek@ans-gniezno.edu.pl</i>
13	Imię i nazwisko koordynatora(ów) przedmiotu, stopień lub tytuł naukowy, adres e-mail	mgr inż. Mateusz Leszek <i>m.leszek@ans-gniezno.edu.pl</i>
14	Język wykładowy	polski
15	Tryb prowadzenia zajęć	stacjonarny
16	Sposób prowadzenia zajęć	-
17	Narzędzia informatyczne wykorzystywane do prowadzenia zajęć, udostępniania materiałów i komunikacji ze studentami	Microsoft Teams / platforma Moodle
15	Przedmioty wprowadzające	- Teoretyczne podstawy informatyki, - Wprowadzenie do programowania,
16	Wymagania wstępne	1. Podstawowa wiedza o budowie i zasadzie działania systemu komputerowego. 2. Podstawowa znajomość składni i struktur dowolnego języka programowania. 3. Podstawowa wiedza o budowaniu prostych algorytmów.
17	Cele przedmiotu:	
C1	Przekazanie studentom podstawowej wiedzy dotyczącej budowy i zasady działania systemów komputerowych.	
C2	Przedstawienie podstawowych zasad programowania niskopoziomowego w języku assembler dla architektury IA-32.	
C3	Nauka implementacji prostych problemów programistycznych z wykorzystaniem języka assembler i kompilatora NASM.	
18	Forma zajęć, liczba godzin wymagająca bezpośredniego udziału nauczyciela akademickiego, liczba godzin nakładu pracy studenta	

Forma zajęć		Liczba godzin
1. wykład		30
2. laboratorium		15
Suma godzin		45
L.p.	Całkowity nakład pracy studenta	
1	Nakład pracy związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela akademickiego wynosi:	Godzinowe obciążenie studenta
	wykład: 30 godz.	47 godzin
	laboratorium: 15 godz.	
	udział w egzaminie: 2 godz.	
Nakład pracy związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela akademickiego wynosi 47 godzin , co odpowiada 1,5 punktom ECTS .		
2	<p>Bilans nakładu pracy studenta:</p> <ul style="list-style-type: none"> przygotowanie do laboratoriów: 5 godzin, samodzielne studiowanie tematyki: 15 godzin, przygotowanie do egzaminu: 10 godzin, przygotowanie do zaliczenia z laboratoriów: 5 godzin, <p>Łączny nakład pracy studenta wynosi 35 godzin, co odpowiada 1,5 punktom ECTS.</p>	35 godzin
3	Łączny nakład pracy studenta (pozycja 1+2)	82 godzin
4	Punkty ECTS za przedmiot	3 ECTS
5	Liczba punktów ECTS, którą student musi osiągnąć w ramach zajęć o charakterze praktycznym w tym zajęć laboratoryjnych, warsztatowych, projektowych	1 ECTS
Efekty uczenia się - wiedza	K_W10: student ma wiedzę z zakresu działania systemu komputerowego, arytmetyki maszyn cyfrowych, organizacji komputera na poziomie przesłań międzyrejestrów i poziomie instrukcji maszynowych; zna organizację i architektury systemów pamięci oraz działanie urządzeń wejścia/wyjścia i sposobu ich komunikacji w systemie komputerowym.	
	K_W13: student ma poszerzoną wiedzę w zakresie logicznej i fizycznej budowy komputera i urządzeń otoczenia procesora oraz pogłębione podstawy teoretyczne działania jego podstawowych elementów składowych (pamięć, CPU).	
	K_W18: ma wiedzę o cyklu życia sprzętu komputerowego, stanie oraz najnowszych trendach i uwarunkowaniach rozwojowych informatyki w odniesieniu do architektury komputerów.	
Efekty uczenia się - umiejętności	K_U01: student potrafi samodzielnie pozyskiwać informacje z literatury, baz danych i dokumentacji technicznych oraz efektywnie pozyskiwać wiedzę, w tym w systemie kształcenia zdalnego (blended/e-learning) w zakresie architektury komputerów	
	K_U09: student ma umiejętność pisania prostych programów na poziomie assemblera z użyciem instrukcji warunkowych, pętli, operacji na liczbach całkowitych i ciągach znaków, tablic oraz stosu.	
	K_U13: student potrafi pisać, uruchamiać, śledzić i testować programy środowiska programistycznego dla języka assembler notacji (NASM) wykorzystując znajomość paradygmatów i metod programowania niskopoziomowego dla architektury IA-32.	
Efekty uczenia się – kompetencje społeczne	K_K01: student rozumie potrzebę i zna możliwości ciągłego dokształcania się, krytycznie odnosi się do posiadanej wiedzy, podnoszenia kompetencji zawodowych, osobistych i społecznych	
	K_K02: student ma świadomość ważności i rozumie pozatechniczne aspekty i skutki działalności inżyniera-informatyka, w tym jej wpływ na środowisko i związaną z tym odpowiedzialność za podejmowane decyzje; dba o dobre tradycje zawodu informatyka.	

2. TREŚCI PROGRAMOWE ODNIESIONE DO EFEKTÓW UCZENIA SIĘ

Treści programowe		liczba godzin
Forma: wykład		
W1	Podstawowe pojęcia: informacja cyfrowa, Model von Neumana, ALU	3
W2	Pamięci półprzewodnikowe	3
W3	Procesor - część I - ALU, CPU, Cykl rozkazowy, RISC i CISC	3
W4	Procesor - część II - Stos, Pamięć CACHE, Przetwarzanie potokowe i równoległe, HT i procesory wielordzeniowe	3
W5	Procesor - część III - Stronicowanie pamięci, tryby pracy procesora, rozszerzenia multimedialne, rozwój procesorów	3
W6	Układy we/wy - jak je widzi procesor? System przerwań, DMA	3
W7	Układy otoczenia procesora - część I - Chipset; CPU vs. APU	3
W8	Układy otoczenia procesora - część II - Magistrale kart rozszerzeń	3
W9	Budowa i funkcje BIOS-u (UEFI); Standardy płyt głównych; Zasilacze	1,5
W10	Transmisja asynchroniczna; RS-232; USB	1,5
W11	Pamięci masowe; Modemy	1,5
W12	Systemy wbudowane	1,5
Forma: laboratoria		
L1	Wprowadzenie do asemblera i środowiska NASM	3
L2	Operatory logiczne i struktury sterujące	3
L3	Tablice i stos	2
L4	Ciągi znaków	2
L5	Operacje konwersji danych	2
L6	Funkcje zewnętrzne	2

3. Literatura

Literatura podstawowa	<ol style="list-style-type: none"> 1. Metzger P., Anatomia PC, wyd. IX, Helion, 2004 - https://biblioteka.ans-gniezno.edu.pl/?det&oid=107&dt=0 2. Dokumentacja architektury IA-32: https://www.intel.com/content/www/us/en/developer/articles/technical/intel-sdm.html 3. Dokumentacja kompilatora NASM: https://www.nasm.us/doc/
Literatura uzupełniająca	<ol style="list-style-type: none"> 1. Stallings W., SYSTEMY OPERACYJNE Architektura, funkcjonowanie i projektowanie, wyd. IX, Helion, 2018 - https://biblioteka.ans-gniezno.edu.pl/?det&oid=9613&dt=0 2. Kruk S., Asembler : wykłady i ćwiczenia, Mikom, 2003 - https://biblioteka.ans-gniezno.edu.pl/?det&oid=537&dt=1&hdg=-1

4. Metody dydaktyczne

Forma:	Metody dydaktyczne:
wykład	wykład informacyjny – prezentacja multimedialna, opowiadanie.
laboratoria	metody ćwiczeniowo - praktyczne, metoda doświadczeniowa – prezentacja multimedialna.

5. Metody i kryteria oceniania

Forma zajęć: wykład	Forma zaliczenia: egzamin
Uzyskane punkty są przeliczane na oceny według następującej skali: Procent punktów: Ocena: 91-100% Bardzo dobry 85-90% Dobry plus 76-84% Dobry 66-75% Dostateczny plus 51-65% Dostateczny 0-50% Niedostateczny	
Opis: Egzamin w formie testu jednokrotnego wyboru na platformie Moodle.	
Forma zajęć: laboratorium	Forma zaliczenia: zaliczenie z oceną
Uzyskane punkty są przeliczane na oceny według następującej skali: Procent punktów: Ocena: 91-100% Bardzo dobry 85-90% Dobry plus 76-84% Dobry 66-75% Dostateczny plus 51-65% Dostateczny 0-50% Niedostateczny	
Opis: Zaliczenie w formie pisemnej, rozwiązanie przedstawionego algorytmu w języku assembler (NASM– IA32).	
Warunkiem zaliczenia przedmiotu jest uzyskanie oceny pozytywnej ze wszystkich form zajęć.	

	Zatwierdzenie karty opisu zajęć	
	Stanowisko Tytuł/stopień naukowy, imię nazwisko	Podpis
Opracował	mgr inż. Mateusz Leszek	
Zatwierdził	Dyrektor Instytutu Nauk Technicznych	