

KARTA PRZEDMIOTU (SYLABUS)

Opis przedmiotu

| | | | |
|---|--|--|---------------------|
| Kod przedmiotu | Nazwa przedmiotu | PROGRAMOWANIE NISKOPOZIOMOWE | |
| IT/P/1/ST/B1-8 | | LOW-LEVEL PROGRAMMING | |
| Język wykładowy | język polski | | |
| Rok akademicki | 2024/2025 | | |
| Kierunek | Informatyka techniczna | | |
| w zakresie | - | | |
| Poziom studiów | studia pierwszego stopnia | | |
| Profil studiów | praktyczny | | |
| Forma studiów | studia stacjonarne | | |
| Semestr / semestry | 3 | | |
| Przynależność do grupy zajęć | B1. Grupa zajęć kierunkowych - obowiązkowych | | |
| Status przedmiotu | obowiązkowy | | |
| Formy realizacji zajęć dydaktycznych, wymiar, punkty ECTS | Forma zajęć | Liczba godzin zajęć dydaktycznych | Liczba punktów ECTS |
| | Wykład | 30 [h] | 5 ECTS |
| | Laboratorium | 30 [h] | |
| Powiązanie przedmiotu | z profilem studiów | związany z kształtowaniem umiejętności praktycznych | 2,5 ECTS |
| | z uprawnieniami | służy do zdobywania przez studenta kompetencji inżynierskich | 4 ECTS |
| | z dyscypliną | informatyka techniczna i telekomunikacja | 5 ECTS |
| Forma nauczania | tradycyjna – zajęcia zorganizowane w Uczelni i/lub zajęcia z wykorzystaniem metod i technik kształcenia na odległość (max. 1,2 ECTS) | | |
| Wymagania wstępne | Wymagana znajomość przedmiotów: podstawy programowania, teoretyczne podstawy informatyki | | |
| Jednostka prowadząca | Katedra Informatyki i Teleinformatyki | | |
| Koordynator | dr Artur Hermanowicz | | |
| Adres strony internetowej pjo | www.wteii.uniwersytetradom.pl | | |
| Adres e-mail, telefon koordynatora | artur.hermanowicz@urad.edu.pl, +48 48 361 78 21 | | |

EFEKTY UCZENIA SIĘ, TREŚCI PROGRAMOWE, REALIZACJA ZAJĘĆ DYDAKTYCZNYCH, WERYFIKACJA EFEKTÓW UCZENIA SIĘ

| | |
|--------------------|--|
| Cel kształcenia: | Poznanie metod przechowywania i przetwarzania informacji w bliskim związku ze sprzętem komputerowym, a w szczególności: poznanie języka assembler, programowanie elementów systemu komputerowego, efektywniejsze wykorzystanie komputera dzięki znajomości zasad jego działania. |
| Treści programowe: | <p>Wykład [W1]:</p> <ol style="list-style-type: none"> 1. Język maszynowy, komputerowa reprezentacja informacji. 2. Język assembler: etapy tworzenia programu w języku assembler, budowa programu. Podstawowe rozkazy procesorów 80x86: rozkazy przesłań, rozkazy arytmetyczne i logiczne, skoki warunkowe i bezwarunkowe, tworzenie pętli programowych, tworzenie procedur. 3. Korzystanie z funkcji systemu operacyjnego, przerwania i ich obsługa, bezpośrednie działanie na pamięci karty graficznej w trybie znakowym i graficznym. 4. Zalety i wady programowania niskopoziomowego, porównanie programów w assemblerze do programów w językach wysokiego poziomu. 5. Łączenie assemblera z językami wysokiego poziomu, wstawki assemblerowe, wskaźniki w C, bezpośrednie działanie na pamięci. <p style="text-align: right;">Suma: 30 [h]</p> <p>Laboratorium [PP, U1, U2, K1]:</p> <ol style="list-style-type: none"> 1. Zapoznanie z podstawowymi instrukcjami języka assembler dla procesorów x86. Współpraca assemblera z językami wysokiego poziomu – tworzenie wstawek assemblerowych. 2. Programowanie prostych wstawek assemblerowych wykorzystujących rozkazy przesłań, instrukcje arytmetyczne i logiczne, skoki, tworzenie pętli. 3. Tworzenie całego programu w assemblerze. Wykorzystanie assemblera, linkera i debuggera. Implementacje prostych algorytmów w assemblerze. 4. Tworzenie procedur. Implementacje programów wykorzystujących podprogramy. |

| | |
|--|---|
| | <p>5. Wykorzystanie przerwań. Zastosowanie przerwań do obsługi wejścia i wyjścia tekstowego.</p> <p>6. Zastosowanie asemblera do bezpośredniego działania na pamięci karty graficznej. Implementacje prostych algorytmów dotyczących grafiki komputerowej, kreślenie linii, okręgu itp.</p> <p>7. Programowanie koprocatora. Zastosowanie koprocatora do obliczeń zmiennoprzecinkowych. Współpraca procesora z koprocotorem.</p> <p style="text-align: right;">Suma: 30 [h]</p> |
| Metody dydaktyczne (kształcenia): | <p>– metody podające (wykład informacyjny),</p> <p>– metody problemowe (wykład problemowy, wykład konwersatoryjny),</p> <p>– metody aktywizujące (dyskusja dydaktyczna),</p> <p>– metody eksponujące (film, ekspozycja, pokaz),</p> <p>– metody programowane (z wykorzystaniem komputera),</p> <p>– metody praktyczne (pokaz, ćwiczenia laboratoryjne, rachunkowe, symulacja)</p> |
| Rygor zaliczenia, kryteria oceny osiągniętych efektów uczenia się, sposób obliczania oceny końcowej: | <p>Warunkiem zaliczenia przedmiotu jest osiągnięcie wszystkich wymaganych efektów kształcenia określonych dla danego przedmiotu. Uzyskanie pozytywnych ocen ze wszystkich form zajęć wchodzących w skład danego przedmiotu jest równoznaczne z jego zaliczeniem i zdobyciem przez studenta liczby punktów ECTS przyporządkowanej temu przedmiotowi. Sposób obliczenia oceny końcowej z przedmiotu określony został w regulaminie studiów.</p> <p>Sposób obliczania oceny z poszczególnych form zajęć przedstawia się następująco:</p> <p>Na ocenę z laboratorium składa się: punktowa ocena sprawdzianów pisemnych oraz praktycznych przy komputerze (90%), punktowa ocena aktywności na zajęciach (10%).</p> <p>Ocena z zaliczenia wykładu – wynik otwartego testu pisemnego.</p> <p>Zdobyte w poszczególnych formach zajęć punkty przeliczane zostają na ocenę wg skali:</p> <p>Ocena 2 poniżej 51%</p> <p>Ocena 3 od 51%</p> <p>Ocena 3,5 od 61%</p> <p>Ocena 4 od 71%</p> <p>Ocena 4,5 od 81%</p> <p>Ocena 5 od 91%</p> |

| Efekty uczenia się dla przedmiotu w odniesieniu do efektów kierunkowych i formy zajęć | | | | Metody weryfikacji efektów uczenia się | |
|---|--|------------------------------------|--------------|--|---------------------------------------|
| Numer efektu uczenia się | Opis efektów uczenia się dla przedmiotu (PEU) Student, który zaliczył przedmiot (W) zna i rozumie/ (U) potrafi /(K) jest gotów do: | Kierunkowy efekt uczenia się (KEU) | Forma zajęć | Forma weryfikacji (zaliczeń) | Metody sprawdzania i oceny |
| W1 | budowę procesora oraz metody programowania go na poziomie listy rozkazów, funkcje systemu operacyjnego oraz możliwości ich wykorzystania w programowaniu niskopoziomowym, zalety i wady programowania niskopoziomowego oraz różnice w porównaniu do języków wysokiego poziomu, możliwości wykorzystania kodu niskopoziomowego w językach wysokiego poziomu | K_WG07 K_WG09 | wykład | zaliczenie | pisemny test otwarty |
| U1 | zaprogramować niskopoziomowo poszczególne elementy systemu komputerowego rozumiejąc ich budowę i zasady działania, sformułować algorytm i zaprogramować go w języku niskiego poziomu posługując się odpowiednimi narzędziami, wykorzystać funkcje systemu operacyjnego odpowiednie do rozwiązania postawionego zadania | K_UW02 K_UW12 | laboratorium | zaliczenie | sprawdzian praktyczny przy komputerze |
| U2 | pozyskiwać informacje z literatury technicznej dotyczące zagadnień związanych z funkcjonowaniem sprzętu komputerowego oraz formułować wnioski | K_UW01 K_UU26 | laboratorium | zaliczenie | sprawdzian praktyczny przy komputerze |

| | | | | | |
|----|---|------------------|-----------------------|------------|--|
| K1 | obserwacji rozwoju technologii komputerowych będąc świadomym konieczności aktualizowania i poszerzania wiedzy w zakresie nauk informatycznych | K_KK01 K_KK02 | wykład / laboratorium | obserwacja | dyskusja, aktywność na zajęciach, prezentacja wyników prac |
|----|---|------------------|-----------------------|------------|--|

Literatura i pomoce naukowe

1. Farbaniec D.: Asembler. Programowanie, Helion, Gliwice 2019
2. Irvine K.R.: Asembler dla procesorów Intel, Helion, Gliwice 2003
3. Kowalczyk A.: Assembler, Croma, Wrocław 1999
4. Kruk S.: Asembler. Wykłady i ćwiczenia, Mikom, Warszawa 2003
5. Van Hoey J.: Programowanie w asemblerze x64. Od nowicjusza do znawcy AVX, Helion, Gliwice 2023
6. Wróbel E.: Praktyczny kurs asemblera, wyd. 2, Helion, Gliwice 2011

Nakład pracy studenta potrzebny do osiągnięcia zakładanych efektów uczenia się – bilans punktów ECTS

| Udział w zajęciach, aktywność | Obciążenie studenta [h] | | |
|---|-----------------------------|---|---------------------|
| | Inne godz. kontaktowe (IGK) | Zajęcia bez nauczyciela-praca własna studenta (ZBN) | Zajęcia dydaktyczne |
| Udział w wykładach | X | X | 30 [h] |
| Udział w ćwiczeniach / laboratoriach / projektach / seminariach | X | X | 30 [h] |
| Udział w konsultacjach | 6 [h] | X | X |
| Przygotowanie do wykładów / ćwiczeń / laboratoriów / projektów / seminariów | X | 59 [h] | X |
| Przygotowanie do zaliczenia/egzaminu | | | |
| Sumaryczne obciążenie pracą studenta | 6 [h] / 0,2 ECTS | 59 [h] / 2,4 ECTS | 60 [h] / 2,4 ECTS |
| Punkty ECTS za przedmiot | 5 ECTS | | |

Informacje dodatkowe, uwagi

W przypadku studentów ze szczególnymi potrzebami, w tym: z niepełnosprawnością, przewlekle chorych, określone powyżej (w karcie) metody i formy weryfikacji efektów uczenia się dostosowuje się odpowiednio do indywidualnych potrzeb tych studentów. Szczegółowe zasady i formy wsparcia studentów ze szczególnymi potrzebami: w tym z niepełnosprawnością, przewlekle chorych podczas zajęć, zaliczeń i egzaminów określono w: Regulaminie Studiów, Zasadach Studiowania, Procedurze dotyczącej zapewnienia dostępności procesu kształcenia studentom ze szczególnymi potrzebami, w tym: z niepełnosprawnością, przewlekle chorych.