

Programowanie strukturalne

Sylabus zajęć

Informacje podstawowe

Kierunek studiów Technologie informatyczne		Cykl dydaktyczny 2024/25
Specjalność -		Kod zajęć 17TINS.31K.07259.24
Jednostka organizacyjna Nadnotecki Instytut UAM w Pile		Języki wykładowe polski
Poziom studiów studia inżynierskie pierwszego stopnia		Obligatoryjność Obowiązkowy
Forma studiów studia stacjonarne		Blok zajęciowy Przedmioty kierunkowe
Profil studiów profil praktyczny		
Koordynator zajęć	Jerzy Szymański	
Prowadzący zajęcia	Jerzy Szymański	
Okres Semestr 1	Forma zajęć / liczba godzin / forma zaliczenia - Laboratorium: 30, Zaliczenie z oceną - Konwersatorium: 30, Egzamin	Liczba punktów ECTS 6

Cele kształcenia dla zajęć

Kod	Cel
C1	Celem przedmiotu jest zapoznanie studentów z podstawami programowania korzystając z paradygmatu programowania strukturalnego.
C2	Nauczenie praktycznego programowania w językach C oraz Python.

Wymagania wstępne

Brak.

Efekty uczenia się dla zajęć

Kod	Efekty uczenia się dla zajęć w zakresie	Efekty uczenia się dla kierunku	Metody weryfikacji osiągnięcia efektów uczenia się dla zajęć
Wiedzy - Student/ka:			
W1	zna podstawowe konstrukcje programistyczne (instrukcje sterujące, wywoływanie procedur i funkcji oraz różne typy przekazywania parametrów)	TIN_K3_W04_inz	Egzamin pisemny
W2	zna podstawowe wbudowane typy danych, a w szczególności różne sposoby reprezentacji danych liczbowych, zasady arytmetyki, ograniczenia i błędy zaokrągleń; rozumie problemy związane z implementacją arytmetyki abstrakcyjnej (matematycznej) w komputerach	TIN_K3_W06_inz	Egzamin pisemny
W3	zna zasady tworzenia złożonych typów danych (tablice, struktury, rekordy/struktury) i potrafi je stosować w algorytmach i programach opisujących sytuacje praktyczne	TIN_K3_W06_inz	Egzamin pisemny
W4	zna zasady dynamicznego zarządzania pamięcią (wskaźniki z typem i bez typu - w tym wskaźniki do funkcji, przydział i zwalnianie pamięci) oraz umie implementować podstawowe dynamiczne struktury danych (kolejka, stos, lista, tablica wskaźników)	TIN_K3_W06_inz	Egzamin pisemny
Umiejętności - Student/ka:			
U1	umie wyrazić wybrane algorytmy w postaci sieci działań, potrafi formułować odpowiednie warunki logiczne i zapisać je za pomocą wyrażeń i operatorów zgodnie ze składnią języka programowania	TIN_K3_U05_inz	Kolokwium pisemne
U2	potrafi czytać i analizować kod proceduralny napisany w językach C oraz Python i rozumie istotę i efekty wykonywanych operacji	TIN_K3_U06	Kolokwium pisemne
U3	Potrafi pisać programy proceduralne w językach oraz Python, uruchamiać je i testować w środowisku Dev-C++ oraz sprawdzarkach automatycznych; potrafi pisać programy wykonujące operacje na plikach	TIN_K3_U06	Kolokwium pisemne
U4	potrafi samodzielnie rozszerzać swoje kompetencje programistyczne.	TIN_K3_U38	Kolokwium pisemne

Treści programowe dla zajęć

Lp.	Treści programowe dla zajęć	Efekty uczenia się dla zajęć	Formy zajęć
1.	Tworzenie oprogramowania. Algorytm Procesor, program, język programowania. Alfabet, składnia i semantyka. Język maszynowy oraz języki wyższego rzędu. Kompilacja, interpretacja i konsolidacja programu. Zmienne, typy, pierwszy program.	U1, U3	Konwersatorium, Laboratorium

Lp.	Treści programowe dla zajęć	Efekty uczenia się dla zajęć	Formy zajęć
2.	Sterowanie przebiegiem programu. Prawda – fałsz – wstępne uwagi o wyrażeniach logicznych. Instrukcje złożone w instrukcjach sterujących. Zakresy ważności nazw, czas życia obiektów, przesłanianie nazw. Instrukcje warunkowa (pełna, niepełna i przełącznik). Pętle for, while i do while. Przerwa i kontynuacja wykonywania pętli. Instrukcja skoku.	W1, U1	Konwersatorium, Laboratorium
3.	Typ danych, typy podstawowe. Reprezentacja danych liczbowych i znakowych. Pojęcie typu. Rola typu w procesie tworzenia programu, stałe i zmienne. Znaczenie typu w procesie kompilacji. Typy całkowite – reprezentacja liczb. Typ znakowy – kodowanie znaków. Typy zmiennoprzecinkowe - reprezentacja. Arytmetyka (całkowita a zmiennoprzecinkowa). Definiowanie stałych różnych typów. Konwersje typów, rzutowania.	W2, U2	Konwersatorium, Laboratorium
4.	Zmienne i wyrażenia. Operatory. Operatory arytmetyczne. Operatory relacyjne. Operatory logiczne. Operatory bitowe. Operator przypisania. Operator rzutowania. Priorytety operatorów. Łączność operatorów.	W2, U1	Konwersatorium, Laboratorium
5.	Złożone typy danych. Tablice i tablice wielowymiarowe. Elementy tablic. Inicjalizacja tablic. Łącuchy znaków (tablice znakowe). Formatowanie wejścia/wyjścia. Struktury (rekordy).	W3, U2	Konwersatorium, Laboratorium
6.	Procedury i funkcje. Pojęcie funkcji; zwracanie wyniku; wartość void. Stos. Przekazywanie parametrów. Prototyp funkcji (typ funkcyjny). Czas życia i zakres ważności nazwy. Zakres lokalny. Zakres pliku. Zmienne statyczne. Zmienne globalne. Ukrywanie informacji. Funkcje biblioteczne.	W1, U2, U4	Konwersatorium, Laboratorium
7.	Przeciążanie operatorów i funkcji. Istota przeciążania. Zagadnienie identyczności typów argumentów. Dopasowanie argumentów do funkcji przeciążonych.	W4	Konwersatorium, Laboratorium
8.	Dynamiczne zarządzanie pamięcią. Wskaźniki. Wskaźniki, referencje i dereferencje. Dynamiczna alokacja pamięci, sarta. Operator sizeof(). Wskaźniki w zastosowaniu do tablic. Wskaźniki w zastosowaniu do przekazywania parametrów. Wskaźniki do struktur. Wskaźniki do funkcji. Wskaźniki bez typu (untyped pointers).	W4, U4	Konwersatorium, Laboratorium

Informacje dodatkowe

Forma zajęć	Metody i formy prowadzenia zajęć
Laboratorium	Rozwiązywanie zadań (np.: obliczeniowych, artystycznych, praktycznych)
Konwersatorium	Wykład z prezentacją multimedialną wybranych zagadnień, Wykład konwersatoryjny

Forma zajęć	Warunki zaliczenia zajęć
Laboratorium	Procentowo: od 90% - bdb od 80% - db+ od 70% - db od 60% - dst+ od 50% - dst poniżej 50% - ndst
Konwersatorium	Procentowo: od 90% - bdb od 80% - db+ od 70% - db od 60% - dst+ od 50% - dst poniżej 50% - ndst

Literatura

Obowiązkowa

1. B. Eckel, Thinking in C, Beta 3 . 2006
2. N. Wirth, Wprowadzenie do programowania systematycznego. Wydawnictwo Naukowo-Techniczne. Warszawa 1978.
3. M. Lutz, Python. Wprowadzenie. Wydanie V, Helion, 2020.

Dodatkowa

1. J. Grębosz, Symfonia C++, Oficyna Kallimach, Kraków 1993.
2. Zanurkuj w Pythonie. <http://pl.wikibooks.org/wiki/Python>

Nakład pracy studenta i punkty ECTS

Rodzaje zajęć studenta	Średnia liczba godzin* przeznaczonych na zrealizowane rodzaje zajęć
Laboratorium	30
Przygotowanie do zajęć	40
Przygotowanie projektu	15
Inne	50
Konwersatorium	30
Łączny nakład pracy studenta	Liczba godzin 165
Liczba punktów ECTS	ECTS 6

* godzina (lekcyjna) oznacza 45 minut

Efekty uczenia się dla kierunku

Kod	Treść
TIN_K3_U05_inz	Absolwent/ka potrafi pisać, uruchamiać i testować programy w wybranym środowisku programistycznym
TIN_K3_U06	Absolwent/ka potrafi czytać ze zrozumieniem programy zapisane w języku programowania imperatywnego
TIN_K3_U38	Absolwent/ka potrafi pogłębiać i aktualizować swoją wiedzę i umiejętności z zakresu informatyki
TIN_K3_W04_inz	Absolwent/ka zna i rozumie podstawowe konstrukcje programistyczne (przypisanie, instrukcje sterujące, wywoływanie podprogramów i przekazywanie parametrów) oraz pojęcia składni i semantyki języków programowania
TIN_K3_W06_inz	Absolwent/ka zna i rozumie podstawowe struktury danych i wykonywane na nich operacje (reprezentacja danych liczbowych, arytmetyka i błędy zaokrągleń, tablice, napisy, zbiory, rekordy, pliki, wskaźniki i referencje, struktury wskaźnikowe, listy, stosy, kolejki, drzewa i grafy)