**Nazwa przedmiotu:**

Podstawy programowania i przetwarzania danych

**Koordynator przedmiotu:**

dr Anna Cena

**Status przedmiotu:**

Obowiązkowy

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Matematyka

**Grupa przedmiotów:**

Wspólne

**Kod przedmiotu:**

1120-MA000-LSP-0115

**Semestr nominalny:**

1 / rok ak. 2020/2021

**Liczba punktów ECTS:**

5

**Liczba godzin pracy studenta związanych z osiągnięciem efektów uczenia się:**

1. godziny kontaktowe – 65 h; w tym
 a) obecność na wykładach – 15 h
 b) obecność na ćwiczeniach – 15 h
 c) obecność na laboratoriach – 30 h
 d) konsultacje – 5 h
2. praca własna studenta – 40 h; w tym
 a) przygotowanie do ćwiczeń i do kolokwiów – 15 h
 b) zapoznanie się z literaturą – 5 h
 c) przygotowanie do laboratoriów – 20 h
Razem 105 h, co odpowiada 5 pkt. ECTS

**Liczba punktów ECTS na zajęciach wymagających bezpośredniego udziału nauczycieli akademickich:**

1. obecność na wykładach – 15 h
2. obecność na ćwiczeniach – 15 h
3. obecność na laboratoriach – 30 h
4. konsultacje – 5 h
Razem 65 h, co odpowiada 2 pkt. ECTS

**Język prowadzenia zajęć:**

polski

**Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym:**

1. obecność na laboratoriach – 30 h
2. przygotowanie do laboratoriów – 20 h
Razem 50 h, co odpowiada 2 pkt. ECTS

**Formy zajęć i ich wymiar w semestrze:**

|  |  |
| --- | --- |
| Wykład:  | 30h |
| Ćwiczenia:  | 15h |
| Laboratorium:  | 30h |
| Projekt:  | 0h |
| Lekcje komputerowe:  | 0h |

**Wymagania wstępne:**

brak

**Limit liczby studentów:**

Bez limitu

**Cel przedmiotu:**

Kurs jest wprowadzeniem do programowania imperatywnego z użytkowymi elementami technik programowania obiektowego na przykładzie języka Python 3. Student poznaje pojęcie algorytmu, funkcji, programu, rekurencji, tablicy (listy), a także najbardziej podstawowe algorytmy i struktury danych, które mogą być wykorzystywane w przetwarzaniu danych, m.in. algorytmy wyszukiwania, sortowania i działania na wektorach oraz macierzach (także w podgrupach generowanych przez zmienne typu czynnikowego) oraz tablice dynamiczne, listy jednokierunkowe i drzewa binarne. Ponadto zapoznaje się z wybranymi funkcjami z biblioteki pakietów dla środowiska Python, np. służącymi do generowania wykresów, liczb pseudolosowych z wybranych rozkładów itp. Nabywa także umiejętności analizy złożoności obliczeniowej i pamięciowej poznanych algorytmów.
Na zajęciach ćwiczeniowych student rozwija umiejętności analizy zagadnień problemowych i tworzenia algorytmów służących do ich rozwiązania z wykorzystaniem poznanych na wykładzie wiadomości teoretycznych.
Na zajęciach laboratoryjnych student uczy się praktycznych umiejętności tworzenia pełnych programów, które są oparte na poznanych algorytmach. Szczególną uwaga zwraca się więc na: implementację programu przy użyciu gotowych, udokumentowanych bibliotek, umiejętność przetestowania programu, jego wykonania na konkretnych danych wejściowych, oraz interpretację otrzymanego wyniku. Na wybranych zajęciach laboratoryjnych student rozwiązuje samodzielnie zadania sprawdzające.

**Treści kształcenia:**

1. Pojęcie problemu obliczeniowego
2. Typy skalarne, podstawowe operatory i ich priorytety
3. Reprezentacja liczb całkowitych, zmiennopozycyjnych oraz znaków drukowanych. Błędy arytmetyki zmiennopozycyjnej
4. Instrukcja warunkowa i pętle. Definiowanie i dokumentowanie własnych funkcji
5. Listy (tablice)
6. Złożoność obliczeniowa i pamięciowa algorytmów, Notacje asymptotyczne: O, Θ, Ω. Przykłady rzędów wielkości funkcji, Szacowanie złożoności różnych algorytmów w problemie wyszukiwania zadanego elementu w liście
7. Referencje do obiektów, Kopiowanie płytkie a głębokie,
8. Problem sortowania i jego zastosowania. Permutacje. Stabilność algorytmów sortowania. Proste algorytmy sortowania przez porównywanie: bąbelkowe, przez wybór i przez wstawianie
9. Rekurencja: Wprowadzenie. Wieże z Hanoi; Fraktale i żółw, Sortowanie przez scalanie jako przykład zastosowania techniki dziel i rządź, Dolne ograniczenie złożoności sortowania przez porównywanie; Sortowanie szybkie,
10. Tablice dynamiczne. Analiza kosztu zamortyzowanego operacji append() i pop()
11. Sortowanie małych liczb naturalnych (np. danych jakościowych lub porządkowych): szufladkowe, przez zliczanie, kubełkowe i pozycyjne (LSD, MSD),
12. Elementy programowania obiektowego: proste klasy, pola i metody, Przeciążanie operatorów (metody specjalne). Klasa DynamicArray
13. Lista jednokierunkowa (z dowiązaniami); Binarne drzewo poszukiwań
14. Rekurencja – spamiętywanie, programowanie dynamiczne; Algorytmy z nawrotami
15. Tablice z haszowaniem. Abstrakcyjny typ danych słownik i zbiór

**Metody oceny:**

Dwa kolokwia;
6-7 laboratoryjnych zadań punktowanych;
Do zdobycia maks. 100 p. Ocena końcowa wynika z sumy punktów; ≤50 p. - 2,0; (50,60] – 3,0; (60,70] – 3,5; (70,80] – 4,0; (80,90] – 4,5; >90 – 5,0.
Szczegółowy regulamin zaliczenia podawany jest na początku semestru.

**Egzamin:**

nie

**Literatura:**

1. Harel D., Feldman Y., Rzecz o istocie informatyki. Algorytmika (Algorithmics: The Spirit of Computing), WNT, 2008.
2. Wirth N., Algorytmy + struktury danych = programy (Algorithms + Data Structures = Programs), WNT, 2004.
3. Cormen T.H., Leiserson C.E., Rivest R.L., Stein C., Wprowadzenie do algorytmów (Introduction to Algorithms), PWN, 2017.
4. Bentley J., Perełki programowania (Programming Pearls), Helion, 2012.

**Witryna www przedmiotu:**

brak

**Uwagi:**

## Charakterystyki przedmiotowe

### Profil ogólnoakademicki - wiedza

**Charakterystyka W01:**

Zna podstawy programowania w języku Python (w tym konstrukcje programistyczne: operacje przypisania, pętle, wyrażenia warunkowe, funkcje) i najważniejsze typy danych (w tym skalary, listy, słowniki, zbiory, wektory i macierze) oraz pojęcie problemu i algorytmu.

Weryfikacja:

kolokwia, zadania punktowane na laboratoriach

**Powiązane charakterystyki kierunkowe:** M1\_W21

**Powiązane charakterystyki obszarowe:**

**Charakterystyka W02:**

Zna proste metody i algorytmy wykorzystywane w przetwarzaniu danych

Weryfikacja:

kolokwia, zadania punktowane na laboratoriach

**Powiązane charakterystyki kierunkowe:** M1\_W20

**Powiązane charakterystyki obszarowe:**

### Profil ogólnoakademicki - umiejętności

**Charakterystyka U01:**

Potrafi implementować proste algorytmy w postaci funkcji oraz oceniać ich złożoność obliczeniową i pamięciową oraz stosować je do konstrukcji prostych programów

Weryfikacja:

kolokwia, zadania punktowane na laboratoriach

**Powiązane charakterystyki kierunkowe:** M1\_U18

**Powiązane charakterystyki obszarowe:**

**Charakterystyka U02:**

Umie stosować wybrane narzędzia zaimplementowane w pakietach dla środowiska Python 3, w tym funkcje do tworzenia wykresów oraz wybrane operacje na wektorach i macierzach

Weryfikacja:

kolokwia, zadania punktowane na laboratoriach

**Powiązane charakterystyki kierunkowe:** M1\_U19

**Powiązane charakterystyki obszarowe:**

### Profil ogólnoakademicki - kompetencje społeczne

**Charakterystyka K01:**

Zna potrzebę uczenia się przez całe życie i podnoszenia kompetencji zawodowych

Weryfikacja:

kolokwia, zadania punktowane na laboratoriach

**Powiązane charakterystyki kierunkowe:** M1\_K01

**Powiązane charakterystyki obszarowe:**