* + - 1. **KARTA PRZEDMIOTU**

|  |  |
| --- | --- |
| Nazwa przedmiotu | * + - 1. **Programowanie**
 |

**1. Usytuowanie przedmiotu w systemie studiów**

|  |  |
| --- | --- |
| 1.1. Kierunek studiów  | **Informatyka** |
| 1.2. Forma i ścieżka studiów | **Stacjonarne/Niestacjonarne** |
| 1.3. Poziom kształcenia | **Studia I stopnia** |
| 1.4. Profil studiów | **Praktyczny** |

|  |  |
| --- | --- |
| 1.5. Specjalność | **-** |
| 1.6. Koordynator przedmiotu |  |

**2. Ogólna charakterystyka przedmiotu**

|  |  |
| --- | --- |
| 2.1. Przynależność do grupy przedmiotu | **Kierunkowy/praktyczny** |
| 2.2. Liczba ECTS | **4** |
| 2.3. Język wykładów | **Polski** |
| 2.4. Semestry, na których realizowany jest przedmiot | **III** |
| 2.5.Kryterium doboru uczestników zajęć | **-** |

1. **efekty uczenia się i sposób prowadzenia zajęć**
	1. **Cele przedmiotu**

|  |  |
| --- | --- |
| **Lp.** | **Cele przedmiotu** |
|
| C1 | Dostarczenie studentowi wiedzy dotyczących konstrukcji programistycznych za pomocą, których może on pozyskiwać, analizować i przetwarzać dane. |
| C2 | Uzyskanie przez studentów praktycznej wiedzy i umiejętności z zakresu algorytmów i programowania w języku Python i C++. |
| C3 | Nabycie umiejętności wykorzystania paradygmatów programowania do tworzenia programów komputerowych, szczególnie paradygmatu obiektowego. |
| C4 | Rozwinięcie myślenia abstrakcyjnego oraz myślenia programistycznego. |

* 1. **Przedmiotowe efekty uczenia się, z podziałem na wiedzę, umiejętności i kompetencje, wraz z odniesieniem do kierunkowych efektów uczenia się**

|  |  |  |  |
| --- | --- | --- | --- |
| **Lp.** | **Opis przedmiotowych efektów uczenia się** | **Odniesienie do kierunkowych efektów****uczenia się (symbole)** | **Sposób realizacji (zaznaczyć „X”)** |
| **ST** | **NST** |
| **Zajęcia na Uczelni** | **Zajęcia na platformie** | **Zajęcia na Uczelni** | **Zajęcia na platformie** |
| Po zaliczeniu przedmiotu student w zakresie **wiedzy** zna i rozumie |
| W1 | składnię i semantykę języka C++ i Python w zakresie obiektowości, kolekcji, wykorzystania plików, utrwalania danych w systemie komputerowym, wyjątków, interakcji programu z użytkownikiem, jak również mechanizmów paradygmatu strukturalnego obecnych w tych językach | INF\_W08INF\_W20 | X |  |  | X |
| W2 | konstrukcje programistyczne dla struktur danych w językach C++ i Python oraz sposób ich zastosowania w praktyce | X |  |  | X |
| W3 | istotę i przebieg opracowania algorytmu stanowiącego rozwiązanie problemu programistycznego | X |  |  | X |
| Po zaliczeniu przedmiotu student w zakresie **umiejętności** potrafi |
| U1 | konstruować programy w języku Python i C++ wykorzystując mechanizmy paradygmatu strukturalnego obecne w tych językach | INF\_U15 INF\_U17 INF\_U19 | X |  | X |  |
| U2 | programować obiektowo w języku Python i C++, stosując cechy obiektowości | X |  | X |  |
| U3 | korzystać z bibliotek oraz czytać dokumentację techniczną, celem opracowania programów w języku Python i C++, realizujących operacje na danych oraz wchodzących w interakcję z użytkownikiem | X |  | X |  |
| Po zaliczeniu przedmiotu student w zakresie **kompetencji społecznych** jest gotów do |
| K1 | podzielenia problemu programistycznego na zadania (etapy), a następnie określenia priorytetów tych zadań przed przystąpieniem do ich realizacji | INF\_K04 | X |  | X |  |

**3.3. Formy zajęć dydaktycznych i ich wymiar godzinowy - Studia stacjonarne (ST), Studia niestacjonarne (NST)**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Ścieżka** | **Wykład** | **Ćwiczenia** | **Projekt** | **Warsztat** | **Laboratorium** | **Seminarium** | **Lektorat** | **Zajęcia prowadzone z wykorzystaniem metod i technik kształcenia na odległość w formie wykładu**  | **Inne** | **Punkty ECTS** |
| **ST** |  |  |  |  | 40 |  |  | 30 |  | 4 |
| **NST** |  |  |  |  | 15 |  |  | 10 |  | 4 |

**3.4. Treści kształcenia** (oddzielnie dla każdej formy zajęć: (W, ĆW, PROJ, WAR, LAB, LEK, INNE). Należy zaznaczyć (X), w jaki sposób będą realizowane dane treści (zajęcia na uczelni lub zajęcia na platformie e-learningowej prowadzone z wykorzystaniem metod i technik kształcenia na odległość)

**RODZAJ ZAJĘĆ: WYKŁAD**

|  |  |  |  |
| --- | --- | --- | --- |
| **Lp.** | **Treść zajęć** | **Odniesienie do przedmiotowych efektów uczenia się** | **Sposób realizacji (zaznaczyć „X”)** |
| **ST** | **NST** |
| **Zajęcia na Uczelni** | **Zajęcia na platformie** | **Zajęcia na Uczelni** | **Zajęcia na platformie** |
| **1.** | Wprowadzenie do programowania. Zapoznanie ze środowiskami programistycznymi. Omówienie procesu zaliczenia przedmiotu. | **W3** | **X** |  |  | **X** |
| **2.** | Mechanizmy strukturalne obecne w językach C++ i Python. Typy danych, funkcje, operatory, kolejność działań, instrukcje sterujące, import bibliotek. | **W1** | **X** |  |  | **X** |
| **3.** | Tablice i struktury danych w językach C++ i Python. Przekazywanie tablic i struktur do funkcji. Pojęcie wskaźnika do obszaru pamięci. | **W1, W2** | **X** |  |  | **X** |
| **4.** | Klasy i obiekty w językach Python i C++. Definiowanie atrybutów i metod klas. Tworzenie obiektów klas. | **W1, W2** | **X** |  |  | **X** |
| **5.** | Kolekcje obiektów w językach Python i C++. Projektowanie i implementacja algorytmów działających na kolekcjach obiektów. | **W1, W2, W3** | **X** |  |  | **X** |
| **6.** | Operacje na łańcuchach znaków w Python i C++. | **W1, W2** | **X** |  |  | **X** |
| **7.** | Programy utrwalające dane w Python i C++. | **W1, W2, W3** | **X** |  |  | **X** |
| **8.** | Interakcja między programem w Python i C++, a jego użytkownikiem. | **W1, W2, W3** | **X** |  |  | **X** |
| **9.** | Podsumowanie zajęć i omówienie ocen. |  | **X** |  |  | **X** |

**RODZAJ ZAJĘĆ: LABORATORIUM**

|  |  |  |  |
| --- | --- | --- | --- |
| **Lp.** | **Treść zajęć** | **Odniesienie do przedmiotowych efektów uczenia się** | **Sposób realizacji (zaznaczyć „X”)** |
| **ST** | **NST** |
| **Zajęcia na Uczelni** | **Zajęcia na platformie** | **Zajęcia na Uczelni** | **Zajęcia na platformie** |
| **1.** | Wprowadzenie do tematyki zajęć – języki, narzędzia, sposób uzyskania zaliczenia. | **U1-U3, K1** | **X** |  | **X** |  |
| **2.** | Pisanie programów wykorzystujących mechanizmy strukturalne obecne w językach C++ i Python, z uwzględnieniem funkcji, operatorów, kolejności działań, instrukcji sterujących, importów bibliotek. | **U1, U3, K1** | **X** |  | **X** |  |
| **3.** | Pisanie programów wykorzystujących tablice i struktury danych w językach C++ i Python. | **U1, U3, K1** | **X** |  | **X** |  |
| **4.** | Pisanie programów wykorzystujących klasy użytkownika i ich obiekty w językach Python i C++. | **U1-U3, K1** | **X** |  | **X** |  |
| **5.** | Pisanie programów operujących na kolekcjach obiektów w językach Python i C++. Praktyczne aspekty algorytmizacji i refaktoryzacji kodu. | **U1-U3, K1** | **X** |  | **X** |  |
| **6.** | Pisanie programów operujących na łańcuchach znaków w Python i C++. | **U1-U3, K1** | **X** |  | **X** |  |
| **7.** | Pisanie programów utrwalających dane w Python i C++. | **U1-U3, K1** | **X** |  | **X** |  |
| **8.** | Implementacja interakcji między programem w Python i C++, a jego użytkownikiem. | **U1-U3, K1** | **X** |  | **X** |  |
| **9.** | Podsumowanie zajęć i omówienie ocen. |  | **X** |  | **X** |  |

**3.5. Metody weryfikacji efektów uczenia się** (wskazanie i opisanie metod prowadzenia zajęć oraz weryfikacji osiągnięcia efektów uczenia się oraz sposobu dokumentacji)

Metody weryfikacji:

Wykład – praca pisemna – 100% oceny końcowej.

Laboratorium – kolokwium – zadania praktyczne do wykonania w określonym czasie – 100% oceny końcowej.

|  |  |  |  |
| --- | --- | --- | --- |
| **Efekty przedmiotowe** | **Metody dydaktyczne** | **Metody weryfikacji efektów uczenia się** | **Sposoby dokumentacji** |
| **WIEDZA** |
| **W1-W3** | Prezentacja multimedialna, dyskusja | Praca pisemna zawierająca treści związane z tematyką zawartą w sylabusie | Zarchiwizowana praca pisemna |
| **UMIEJĘTNOŚCI** |
| **U1-U3** | Zajęcia praktyczne z wykorzystaniem stanowisk komputerowych | Kolokwium - zadania praktyczne związane z tematyką zawartą w sylabusie | Zarchiwizowane rozwiązania zadań praktycznych |
| **KOMPETENCJE SPOŁECZNE** |
| **K1** | Zajęcia praktyczne z wykorzystaniem stanowisk komputerowych | Kolokwium - zadania praktyczne związane z tematyką zawartą w sylabusie | Zarchiwizowane rozwiązania zadań praktycznych |

**3.6. Kryteria oceny osiągniętych efektów uczenia się**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Efekt uczenia się** | **Na ocenę 3 lub „zal.”****student zna i rozumie/potrafi/jest gotów do** | **Na ocenę 3,5 student zna i rozumie/potrafi/jest gotów do** | **Na ocenę 4 student zna i rozumie/potrafi/jest gotów do** | **Na ocenę 4,5 student zna i rozumie/potrafi/jest gotów do** | **Na ocenę 5 student zna i rozumie/potrafi/jest gotów do** |
| W | 51-60% wiedzy wskazanej w efektach uczenia się | 61-70% wiedzy wskazanej w efektach uczenia się | 71-80% wiedzy wskazanej w efektach uczenia się | 81-90% wiedzy wskazanej w efektach uczenia się | 91-100% wiedzy wskazanej w efektach uczenia się |
| U | 51-60% umiejętności wskazanych w efektach uczenia się | 61-70% umiejętności wskazanych w efektach uczenia się | 71-80% umiejętności wskazanych w efektach uczenia się | 81-90% umiejętności wskazanych w efektach uczenia się | 91-100% umiejętności wskazanych w efektach uczenia się |
| K | 51-60% umiejętności wskazanych w efektach uczenia się | 61-70% umiejętności wskazanych w efektach uczenia się | 71-80% umiejętności wskazanych w efektach uczenia się | 81-90% umiejętności wskazanych w efektach uczenia się | 91-100% umiejętności wskazanych w efektach uczenia się |

**3.7. Zalecana literatura**

**Podstawowa**

1. Luciano Ramalho, "Zaawansowany Python, wyd. 2. Przejrzyste, zwięzłe i efektywne programowanie", Promise, 2022 – mamy ☺
2. Jerzy Grębosz, "Opus magnum C++. Programowanie w języku C++. Wydanie III poprawione", Helion, 2024 – mamy pierwsze wydanie z 2018

**Uzupełniająca**

1. Matt Weisfeld, "Myślenie obiektowe w programowaniu. Wydanie V", Helion, 2020 – do kupna na naukowa.pl
2. Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, "Projektowanie i analiza algorytmów", Helion, 2003 – nie mamy, nie do kupienia,

proponowane zamienniki:

1. Lech Banachowski, Krzysztof Diks, Wojciech Rytter, „Algorytmy i struktury danych”. Wyd. nowe poprawione PWN. Wydawnictwo Naukowe PWN, 2018.
2. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, „Wprowadzenie do algorytmów”. Wyd. 2. Wydawnictwo Naukowe PWN, 2024.

**4. nakład pracy studenta - bilans punktów ECTS**

|  |  |
| --- | --- |
| **Rodzaje aktywności studenta** | **Obciążenie studenta** |
| **ST** | **NST** |
| **Zajęcia wymagające bezpośredniego kontaktu studenta z nauczycielem akademickim w siedzibie uczelni** | **70** | **25** |
| Zajęcia przewidziane planem studiów | 70 | 25 |
| **Praca własna studenta** | **30** | **75** |
| Przygotowanie bieżące do zajęć, przygotowanie prac projektowych/prezentacji/itp. | 15 | 45 |
| Przygotowanie do zaliczenia zajęć | 15 | 30 |
| **SUMARYCZNE OBCIĄŻENIE GODZINOWE STUDENTA** | **100** | **100** |
| **Liczba punktów ECTS** | **4** | **4** |

|  |  |
| --- | --- |
| Data ostatniej zmiany |  |
| Zmiany wprowadził |  |
| Zmiany zatwierdził |  |