**karta przedmiotu**

1. Podstawowe informacje o przedmiocie

|  |  |
| --- | --- |
| Nazwa przedmiotu | Wstęp do programowania |
| Rocznik studiów | 2022/2023 |
| Kolegium | Kolegium Informatyki Stosowanej |
| Kierunek studiów | Informatyka |
| Poziom kształcenia | Studia pierwszego stopnia |
| Profil kształcenia | Praktyczny |
| Specjalność | - |
| Osoba odpowiedzialna | dr Marek Jaszuk |

1. Wymagania wstępne (wynikające z następstwa przedmiotów)

|  |
| --- |
| Brak |

1. Efekty uczenia się i sposób realizacji zajęć
   1. Cele przedmiotu

|  |  |
| --- | --- |
| C1 | Przekazanie wiedzy w zakresie analizy i opracowania prostych programów komputerowych i implementacji algorytmów |
| C2 | Przekazanie wiedzy w zakresie metodyki i technik programowania oraz rozwiązywania problemów przy pomocy programów komputerowych |
| C3 | Kształtowanie umiejętności wykorzystania zasobów wielokrotnego użycia w konstruowaniu prostych programów komputerowych |
| C4 | Kształtowanie umiejętności projektowania, implementowania, weryfikowania poprawności i debugowania prostych programów w strukturalnym języku programowania, oraz implementowania prostych algorytmów, wraz z oceną ich złożoności |

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

|  |  |  |
| --- | --- | --- |
| Lp. | Opis przedmiotowych efektów uczenia się | Odniesienie do efektów uczenia się dla kierunku |
| Po zaliczeniu przedmiotu student w zakresie **wiedzy** | | |
| P\_W01 | ma wiedzę w zakresie matematyki niezbędną do, analizy i opracowania prostych programów komputerowych oraz implementacji podstawowych algorytmów | K\_W01 |
| P\_W02 | ma uporządkowaną wiedzę w zakresie metodyki i technik programowania, w tym podstawowych technik algorytmicznych oraz znaczenia myślenia algorytmicznego i komputacyjnego w zakresie rozwiązywania problemów przy pomocy programów komputerowych | K\_W03 |
| Po zaliczeniu przedmiotu student w zakresie **umiejętności** | | |
| P\_U01 | potrafi świadomie i skutecznie korzystać z zasobów wielokrotnego użycia w konstruowaniu prostych programów komputerowych | K\_U10 |
| P\_U02 | potrafi zaprojektować, zaimplementować, weryfikować poprawność i debugować proste programy w strukturalnym języku programowania oraz implementować podstawowe algorytmy, a także ocenić ich złożoność | K\_U11 |

* 1. Formy zajęć dydaktycznych oraz wymiar godzin i punktów ECTS

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Studia stacjonarne (ST) | | | | | | | |
| W | K | Ćw | L | ZP | P | eL | ECTS |
| 20 |  |  | 20 |  | 14 |  | 5 |

* 1. Metody realizacji zajęć dydaktycznych

|  |  |
| --- | --- |
| Formy zajęć | Metoda realizacji |
| Wykład | Wykład informacyjny i problemowy, połączony z elementami demonstracji. |
| Laboratorium | Ćwiczenia laboratoryjne przy komputerze. W trakcie zajęć studenci dokonują samodzielnej implementacji i debugowania programów komputerowych z wykorzystaniem środowiska zintegrowanego. |
| Projekt | Projekt polega na samodzielnej implementacji programu rozwiązującego pewien problem programistyczny. Projekt jest realizowany w konsultacji z prowadzącym zajęcia i kończy się omówieniem i prezentacją działającego rozwiązania. |

* 1. Treści kształcenia (oddzielnie dla każdej formy zajęć)

Wykład

|  |  |
| --- | --- |
| Lp. | Treści kształcenia realizowane w ramach wykładów |
| W1 | Wprowadzenie do środowiska pracy w języku Python, zarządzanie pakietami |
| W2 | Zmienne i podstawowe operatory, operacje wejścia-wyjścia |
| W3 | Instrukcje warunkowe i iteracyjne |
| W4 | Listy |
| W5 | Słowniki |
| W6 | Funkcje |
| W7 | Biblioteka Numpy |
| W8 | Wprowadzenie do systemów kontroli wersji |

Laboratorium

|  |  |
| --- | --- |
| Lp. | Treści kształcenia realizowane w ramach laboratorium |
| L1 | Uruchamianie programów w języku Python |
| L2 | Interakcja z programem, wyświetlanie danych, wprowadzanie danych do programu |
| L3 | Proste zadania algorytmiczne z wykorzystaniem instrukcji warunkowych i iteracyjnych |
| L4 | Praca z listami |
| L5 | Praca ze słownikami |
| L6 | Definiowanie funkcji |
| L7 | Wykorzystanie biblioteki Numpy, praca z tablicami |
| L8 | Zarządzanie wersjami kodu z wykorzystaniem systemu kontroli wersji |

Projekt

|  |  |
| --- | --- |
| Lp. | Treści kształcenia realizowane w ramach projektu |
|
| P1 | Opracowanie założeń projektu |
| P2 | Implementacja programu |
| P3 | Prezentacja wyniku |

* 1. Korelacja pomiędzy efektami uczenia się, celami przedmiotu, a treściami kształcenia

|  |  |  |
| --- | --- | --- |
| Efekt uczenia się | Cele przedmiotu | Treści kształcenia |
| P\_W01 | C1 | W1-W8 |
| P\_W02 | C2 | W2-W8 |
| P\_U01 | C3 | L1-L7, P2 |
| P\_U02 | C4 | L1-L7, P1, P3 |

* 1. Metody weryfikacji efektów uczenia się

|  |  |  |
| --- | --- | --- |
| Efekt  uczenia się | Metoda oceny | Forma zajęć, w ramach której następuje weryfikacja efektu |
| P\_W01 | Egzamin w formie testu | W |
| P\_W02 | Egzamin w formie testu | W |
| P\_U01 | Kolokwium z laboratorium | L |
| P\_U02 | Ocena projektu | P |

* 1. Kryteria oceny stopnia osiągnięcia efektów uczenia się

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Efekt uczenia się | Na ocenę 2  student nie potrafi | Na ocenę 3  student potrafi | Na ocenę 4  student potrafi | Na ocenę 5  student potrafi |
| P\_W01 | zademonstrować wiedzy w zakresie matematyki niezbędnej do, analizy i opracowania prostych programów komputerowych oraz implementacji podstawowych algorytmów | zademonstrować na poziomie podstawowym wiedzę w zakresie matematyki niezbędną do, analizy i opracowania prostych programów komputerowych oraz implementacji podstawowych algorytmów | zademonstrować na poziomie dobrym wiedzę w zakresie matematyki niezbędną do, analizy i opracowania prostych programów komputerowych oraz implementacji podstawowych algorytmów | zademonstrować na poziomie bardzo dobrym wiedzę w zakresie matematyki niezbędną do, analizy i opracowania prostych programów komputerowych oraz implementacji podstawowych algorytmów |
| P\_W02 | zademonstrować wiedzy w zakresie metodyki i technik programowania, w tym podstawowych technik algorytmicznych oraz znaczenia myślenia algorytmicznego i komputacyjnego w zakresie rozwiązywania problemów przy pomocy programów komputerowych | zademonstrować na poziomie podstawowym wiedzę w zakresie metodyki i technik programowania, w tym podstawowych technik algorytmicznych oraz znaczenia myślenia algorytmicznego i komputacyjnego w zakresie rozwiązywania problemów przy pomocy programów komputerowych | zademonstrować na poziomie dobrym wiedzę w zakresie metodyki i technik programowania, w tym podstawowych technik algorytmicznych oraz znaczenia myślenia algorytmicznego i komputacyjnego w zakresie rozwiązywania problemów przy pomocy programów komputerowych | zademonstrować na poziomie bardzo dobrym wiedzę w zakresie metodyki i technik programowania, w tym podstawowych technik algorytmicznych oraz znaczenia myślenia algorytmicznego i komputacyjnego w zakresie rozwiązywania problemów przy pomocy programów komputerowych |
| P\_U01 | korzystać z zasobów wielokrotnego użycia w konstruowaniu prostych programów komputerowych | na poziomie podstawowym korzystać z zasobów wielokrotnego użycia w konstruowaniu prostych programów komputerowych | na poziomie dobrym korzystać z zasobów wielokrotnego użycia w konstruowaniu prostych programów komputerowych | na poziomie bardzo dobrym korzystać z zasobów wielokrotnego użycia w konstruowaniu prostych programów komputerowych |
| P\_U02 | zaprojektować i zaimplementować, prostego programu w strukturalnym języku programowania | zaprojektować i zaimplementować, proste programy w strukturalnym języku programowania | zaprojektować, zaimplementować, weryfikować poprawność i debugować proste programy w strukturalnym języku programowania | zaprojektować, zaimplementować, weryfikować poprawność i debugować proste programy w strukturalnym języku programowania oraz implementować podstawowe algorytmy, a także ocenić ich złożoność |

* 1. Literatura

|  |
| --- |
| Literatura podstawowa |
| Luciano Ramalho, Fluent Python: Clear, Concise, and Effective Programming, O'Reilly Media, 2en ed. 2022 ot later |
| Programming Essentials In Python at <http://netacad.com> |
| Qingkai, Kong, Timmy Siauw, Alexandre Bayen, Python Programming And Numerical Methods: A Guide For Engineers And Scientists, https://pythonnumericalmethods.berkeley.edu/ |

|  |
| --- |
| Literatura uzupełniająca |
| R. C. Martin, Clean Code : A Handbook of Agile Software Craftsmanship, Pearson Education, Upper Saddle River, 2009 |
| Michal Jaworski, Tarek Ziadé, Expert Python Programming - Fourth Edition: Master Python by learning the best coding practices and advanced programming concepts, 2021 |
| Mark Lutz, Learning Python: Powerful Object-Oriented Programming, 5 th ed, 2013 or later |

1. Nakład pracy studenta - bilans punktów ECTS

|  |  |
| --- | --- |
| **Rodzaje aktywności** | **Obciążenie studenta** |
| **studia ST** |
| Udział w W/K (UB) | 20 |
| Konsultacje do W/K (UB) | 4 |
| Udział w egzaminie z W (UB) | 2 |
| Samodzielne studiowanie tematyki W/K, w tym przygotowanie do egzaminu/zaliczenia | 16 |
| Udział w C/L (UB) | 20 |
| Konsultacje do C/L (UB) | 14 |
| Samodzielne przygotowanie się do C/L, w tym przygotowanie do zaliczenia | 16 |
| Udział w i konsultacje do PS/PN/eL (UB) | 14 |
| Samodzielne przygotowanie się do zaliczenia PS/PN/eL | 14 |
| **Sumaryczne obciążenie pracą studenta** | **120** |
| **Punkty ECTS za przedmiot** | **5** |
| **Punkty ECTS za zajęcia prowadzone z bezpośrednim udziałem nauczycieli i studentów (UB)** | **3** |
| **Punkty ECTS za zajęcia kształtujące umiejętności praktyczne (PZ)** | **4** |