**karta przedmiotu**

1. Podstawowe informacje o przedmiocie

|  |  |
| --- | --- |
| Nazwa przedmiotu | Programowanie – semestr 2 |
| 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)

|  |
| --- |
| Wstęp do programowania |

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

|  |  |
| --- | --- |
| C1 | Przekazanie wiedzy w zakresie analizy i opracowania programów komputerowych i implementacji algorytmów |
| C2 | Przekazanie wiedzy w zakresie metodyki i technik programowania oraz rozwiązywania problemów przy pomocy programów komputerowych |
| C3 | Budowanie znajomości i zrozumienia metod, narzędzi, teorii i praktyk stosowanych do projektowania, modelowania i implementacji oprogramowania z uwzględnieniem etapów określenia wymagań, specyfikacji, walidacji i testowania oprogramowania |
| C4 | Kształtowanie umiejętności efektywnego wykorzystywania narzędzi stosowanych w konstruowaniu i dokumentacji procesu wytwarzania oprogramowania, ze szczególnym uwzględnieniem narzędzi do kontroli oprogramowania w tym rozproszonego systemu kontroli wersji |
| C5 | Kształtowanie umiejętności wykorzystania zasobów wielokrotnego użycia w konstruowaniu programów komputerowych |
| C6 | Kształtowanie umiejętności projektowania, implementowania, weryfikowania poprawności i debugowania programów w języku programowania wysokiego poziomu, oraz implementowania 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 programów komputerowych oraz implementacji algorytmów | K\_W01 |
| P\_W02 | ma uporządkowaną wiedzę w zakresie metodyki i technik programowania, w tym technik algorytmicznych oraz znaczenia myślenia algorytmicznego i komputacyjnego w zakresie rozwiązywania problemów przy pomocy programów komputerowych | K\_W03 |
| P\_W03 | zna i rozumie metody, narzędzia, teorie i praktyki stosowane do projektowania, modelowania i implementacji oprogramowania z uwzględnieniem etapów określenia wymagań, specyfikacji, walidacji i testowania oprogramowania | K\_W07, K\_W14 |
| Po zaliczeniu przedmiotu student w zakresie **umiejętności** | | |
| P\_U01 | potrafi efektywnie wykorzystywać narzędzia stosowane w konstruowaniu i dokumentacji procesu wytwarzania oprogramowania, ze szczególnym uwzględnieniem narzędzi do kontroli oprogramowania w tym rozproszonego systemu kontroli wersji | K\_U09 |
| P\_U02 | potrafi świadomie i skutecznie korzystać z zasobów wielokrotnego użycia w konstruowaniu programów komputerowych | K\_U10 |
| P\_U03 | potrafi zaprojektować, zaimplementować, weryfikować poprawność i debugować programy w języku programowania wysokiego poziomu oraz implementować 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 |  | 15 |  | 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 | Implementacja programu rozwiązującego zadany problem. Projekt jest realizowany w konsultacji z prowadzącym zajęcia. Na zakończenie semestru studenci prezentują działające rozwiązanie. |

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

Wykład

|  |  |
| --- | --- |
| Lp. | Treści kształcenia realizowane w ramach wykładów |
| W1 | Wprowadzenie do tworzenia aplikacji internetowych z wykorzystaniem HTML5, CSS, JavaScript - środowisko pracy |
| W2 | Język znaczników HTML i arkusze stylów CSS |
| W3 | Obiektowy model dokumentu HTML DOM |
| W4 | Podstawy składni języka JavaScript |
| W5 | Zmienne, tablice i operatory w JavaScript |
| W6 | Funkcje w JavaScript |
| W7 | Formularze HTML |
| W8 | Walidacja formularza w JavaScript |

Laboratorium

|  |  |
| --- | --- |
| Lp. | Treści kształcenia realizowane w ramach laboratorium |
| L1 | Przygotowanie środowiska pracy do tworzenia aplikacji internetowych z wykorzystaniem HTML5, CSS i JavaScript |
| L2 | Tworzenie strony internetowej z wykorzystaniem znaczników HTML i wykorzystanie kaskadowych arkuszy stylów |
| L3 | Wykorzystanie języka JavaScript w stronie internetowej |
| L4 | Obiekty, metody i właściwości w JavaScript |
| L5 | Programowanie w języku JavaScript – przechowywanie danych w zmiennych i tablicach, wykorzystanie operatorów |
| L6 | Wykorzystanie funkcji w JavaScript, węzły, dostęp do elementów, tworzenie list |
| L7 | Tworzenie formularzy w HTML, tworzenie formularza rejestracji |
| L8 | Wykorzystanie JavaScript do walidacji formularza HTML |

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-W9 |
| P\_W02 | C2 | W1-W9 |
| P\_W03 | C3 | W10 |
| P\_U01 | C4 | L10, P2, P3 |
| P\_U02 | C5 | L1-L9, P1, P2 |
| P\_U03 | C6 | L1-L9, 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\_W03 | Egzamin w formie testu | W |
| P\_U01 | Prezentacja projektu | P |
| P\_U02 | Kolokwium z laboratorium | L |
| P\_U03 | Kolokwium z laboratorium | L |

* 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 programów komputerowych | zademonstrować podstawową wiedzę w zakresie matematyki niezbędną do analizy i opracowania programów komputerowych oraz implementacji algorytmów | zademonstrować na poziomie dobrym wiedzę w zakresie matematyki niezbędną do analizy i opracowania programów komputerowych oraz implementacji algorytmów | zademonstrować na poziomie bardzo dobrym wiedzę w zakresie matematyki niezbędną do analizy i opracowania programów komputerowych oraz implementacji algorytmów |
| P\_W02 | zademonstrować wiedzy w zakresie metodyki i technik programowania | zademonstrować wiedzę w zakresie metodyki i technik programowania | zademonstrować wiedzę w zakresie metodyki i technik programowania, w tym technik algorytmicznych | zademonstrować wiedzę w zakresie metodyki i technik programowania, w tym technik algorytmicznych oraz znaczenia myślenia algorytmicznego i komputacyjnego w zakresie rozwiązywania problemów przy pomocy programów komputerowych |
| P\_W03 | zademonstrować znajomości i rozumienia metod, narzędzi, teorii i praktyk stosowanych do projektowania, modelowania i implementacji oprogramowania | na poziomie podstawowym zademonstrować znajomość i rozumienie metod, narzędzi, teorii i praktyk stosowanych do projektowania, modelowania i implementacji oprogramowania z uwzględnieniem etapów określenia wymagań, specyfikacji, walidacji i testowania oprogramowania | na poziomie dobrym zademonstrować znajomość i rozumienie metod, narzędzi, teorii i praktyk stosowanych do projektowania, modelowania i implementacji oprogramowania z uwzględnieniem etapów określenia wymagań, specyfikacji, walidacji i testowania oprogramowania | na poziomie bardzo dobrym zademonstrować znajomość i rozumienie metod, narzędzi, teorii i praktyk stosowanych do projektowania, modelowania i implementacji oprogramowania z uwzględnieniem etapów określenia wymagań, specyfikacji, walidacji i testowania oprogramowania |
| P\_U01 | efektywnie wykorzystywać narzędzi stosowanych w konstruowaniu i dokumentacji procesu wytwarzania oprogramowania, ze szczególnym uwzględnieniem narzędzi do kontroli oprogramowania w tym rozproszonego systemu kontroli wersji | na poziomie podstawowym wykorzystywać narzędzia stosowane w konstruowaniu i dokumentacji procesu wytwarzania oprogramowania, ze szczególnym uwzględnieniem narzędzi do kontroli oprogramowania w tym rozproszonego systemu kontroli wersji | na poziomie dobrym wykorzystywać narzędzia stosowane w konstruowaniu i dokumentacji procesu wytwarzania oprogramowania, ze szczególnym uwzględnieniem narzędzi do kontroli oprogramowania w tym rozproszonego systemu kontroli wersji | na poziomie bardzo dobrym wykorzystywać narzędzia stosowane w konstruowaniu i dokumentacji procesu wytwarzania oprogramowania, ze szczególnym uwzględnieniem narzędzi do kontroli oprogramowania w tym rozproszonego systemu kontroli wersji |
| P\_U02 | korzystać z zasobów wielokrotnego użycia w konstruowaniu programów komputerowych | na poziomie podstawowym korzystać z zasobów wielokrotnego użycia w konstruowaniu programów komputerowych | na poziomie dobrym korzystać z zasobów wielokrotnego użycia w konstruowaniu programów komputerowych | na poziomie bardzo dobrym korzystać z zasobów wielokrotnego użycia w konstruowaniu programów komputerowych |
| P\_U03 | zaprojektować, zaimplementować, weryfikować poprawności i debugować programów w języku programowania wysokiego poziomu | zaprojektować, zaimplementować, programy w języku programowania wysokiego poziomu | zaprojektować, zaimplementować, weryfikować poprawność i debugować programy w języku programowania wysokiego poziomu | zaprojektować, zaimplementować, weryfikować poprawność i debugować programy w języku programowania wysokiego poziomu oraz implementować algorytmy, a także ocenić ich złożoność |

* 1. Literatura

|  |
| --- |
| Literatura podstawowa |
| Kurs JavaScript Essentials 1 na platformie <http://netacad.com> |
| Repozytorium kursów World Wide Web Consortium - <http://www.w3schools.com> |

|  |
| --- |
| Literatura uzupełniająca |
| Ryan Lanciaux , Modern Front-end Architecture: Optimize Your Front-end Development with Components, Storybook, and Mise en Place Philosophy, Apress, 2021 or later |
| Ben Frain, Responsive Web Design with HTML5 and CSS: Develop future-proof responsive websites using the latest HTML5 and CSS techniques, Packt Publishing, 2020, 3rd Edition |

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 |
| Samodzielne studiowanie tematyki W/K, w tym przygotowanie do egzaminu/zaliczenia | 20 |
| Udział w C/L (UB) | 20 |
| Konsultacje do C/L (UB) | 4 |
| Samodzielne przygotowanie się do C/L, w tym przygotowanie do zaliczenia | 36 |
| Udział w i konsultacje do PS/PN/eL (UB) | 15 |
| Samodzielne przygotowanie się do zaliczenia PS/PN/eL | 12 |
| **Sumaryczne obciążenie pracą studenta** | **133** |
| **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)** | **3** |