

## SYLABUS (KARTA PRZEDMIOTU)

Nazwa przedmiotu/modułu (zgodna z zatwierdzonym programem studiów na kierunku)		Punkty ECTS <b>5</b>	Numer katalogowy
<b>Wprowadzenie do programowania</b>			
Nazwa w j. angielskim			
<b>Introduction to programming</b>			
Jednostka(i) realizująca(e) przedmiot/moduł (instytut/katedra)			
<b>Instytut Inżynierii Biosystemów</b>			
Kierownik przedmiotu/modułu			
<b>dr inż. Sebastian Kujawa</b>			
Kierunek studiów	Poziom	Profil	Semestr
<b>Informatyka stosowana</b>	<b>Studia I stopnia</b>	<b>ogólnoakademicki</b>	<b>2</b>
Specjalność	Specjalizacja inżynierska		
-	-		
<b>RODZAJE ZAJĘĆ I ICH WYMIAR GODZINOWY</b> (zajęcia zorganizowane i praca własna studenta)			
Forma studiów: stacjonarne		Forma studiów: niestacjonarne	
- wykłady	<b>15</b>	- wykłady	<b>10</b>
- ćwiczenia laboratoryjne	<b>30</b>	- ćwiczenia laboratoryjne	<b>20</b>
- inne z udziałem nauczyciela	<b>10</b>	- inne z udziałem nauczyciela	<b>10</b>
- praca własna studenta	<b>70</b>	- praca własna studenta	<b>85</b>
Łączna liczba godzin: <b>125</b>		Łączna liczba godzin: <b>125</b>	
<b>CEL PRZEDMIOTU/MODUŁU</b>			
<p>Celem zajęć jest zdobycie wiedzy oraz wykształcenie umiejętności praktycznych z zakresu podstaw programowania. Na zajęciach omawiane są następujące elementy stanowiące wprowadzenie do programowania: systemy liczbowe, algorytmy i schematy blokowe, historia i klasyfikacja języków programowania. Ponadto studenci zapoznają się z podstawowymi zagadnieniami związanymi z programowaniem w języku C, takimi jak: struktura programu, podstawowe typy danych, stałe i zmienne, funkcje służące do pobierania danych z klawiatury, funkcje służące do wyprowadzania informacji na monitor, instrukcje warunkowe, pętle i rekurencja. W ramach ćwiczeń i wykładów są prezentowane i analizowane przykładowe programy w języku C o niewielkim stopniu złożoności.</p>			
<b>METODY DYDAKTYCZNE</b>			
<ol style="list-style-type: none"> <li>1. Wykład.</li> <li>2. Ćwiczenia laboratoryjne, praca indywidualna i w grupie.</li> <li>3. Praca z komputerem.</li> <li>4. Dyskusja na zajęciach.</li> </ol>			
<b>EFEKTY KSZTAŁCENIA</b>			Odniesienie do efektów kierunkowych
Wiedza	<b>E1.</b> Zna i rozumie pojęcie algorytmu oraz podstawowe sposoby prezentacji algorytmów. <b>E2.</b> Opisuje rodzaje języków programowania. <b>E3.</b> Opisuje podstawowe typy danych stosowane w językach programowania. <b>E4.</b> Opisuje działanie podstawowych operatorów programistycznych. <b>E5.</b> Objaśnia działanie instrukcji warunkowych i instrukcji pętli. <b>E6.</b> Objaśnia działanie podstawowych instrukcji wprowadzania i wyprowadzania danych.		<b>IS1A_W16</b> <b>IS1A_W17</b>
Umiejętności	<b>E7.</b> Opracowuje proste algorytmy i prezentuje je z wykorzystaniem schematów blokowych. <b>E8.</b> Tworzy i prezentuje proste programy uwzględniające: pobieranie danych od użytkownika, przetwarzanie danych za pomocą poznanych instrukcji, wyprowadzanie wyników na monitor. <b>E9.</b> Dokonuje deklaracji stałych i zmiennych w tworzonym programie z uwzględnieniem właściwego typu danych. <b>E10.</b> Wyszukuje i eliminuje błędy składniowe w kodzie tworzonego programu.		<b>IS1A_U13</b>
Kompetencje Społeczne	<b>E11.</b> Rozumie potrzebę samodzielnej pracy mającej na celu utrwalenie oraz poszerzenie zdobytej wiedzy i wykształconych umiejętności praktycznych w zakresie wytwarzania aplikacji komputerowych. <b>E12.</b> Potrafi pracować w grupie w celu rozwiązania zadania programistycznego.		<b>IS1A_K01</b> <b>IS1A_K02</b> <b>IS1A_K05</b>

<b>Metody weryfikacji efektów kształcenia</b> Kolokwium pisemne Projekt programistyczny Pisemne zaliczenie wykładów	Numery efektów E1, E3, E4, E5, E6, E7, E8, E9, E11 E8, E9, E10, E11, E12 E2, E3, E4, E5, E6, E8, E9, E11
<b>TREŚCI KSZTAŁCENIA</b>	
<b>Tematyka wykładów:</b> <ul style="list-style-type: none"> <li>• Systemy liczbowe: dziesiętny, dwójkowy, ósemkowy i heksadecymalny.</li> <li>• Algorytmy, ich właściwości i sposoby prezentacji. Schematy blokowe.</li> <li>• Historia i klasyfikacja języków programowania.</li> <li>• Kompilatory i interpretatory. Języki kompilowane a języki interpretowane. Pojęcia: kod źródłowy, plik wynikowy i plik wykonywalny.</li> <li>• Wprowadzenie do języka C. Struktura programu w języku C: funkcje i ich argumenty, funkcja main(), funkcja return(), dołączanie plików nagłówkowych, komentarze.</li> <li>• Podstawowe typy danych: znakowe, łańcuchowe, liczby całkowite i liczby zmiennopozycyjne.</li> <li>• Deklaracja stałych i zmiennych. Dyrektywa #define, modyfikator const. Tablice.</li> <li>• Wyświetlanie informacji na ekranie. Funkcje: puts(), putchar(), printf(), instrukcja cout. Kody sterujące i specyfikatory formatowania.</li> <li>• Wprowadzanie danych z klawiatury. Funkcje: gets(), getchar(), scanf(), instrukcja cin. Operator pobrania adresu.</li> <li>• Przetwarzanie wprowadzonych danych w informację wyjściową – operatory arytmetyczne, wyrażenia, liczniki, operatory zwiększania i zmniejszania, akumulatory, operatory przypisania. Wartości początkowe liczników i akumulatorów.</li> <li>• Instrukcje warunkowe: if, if...else, if...else if...else. Warunki i operatory logiczne. Zagnieżdżenie instrukcji warunkowych.</li> <li>• Pętle i rekurencja. Rodzaje pętli: for, while i do...while. Zagnieżdżenie pętli.</li> </ul> <b>Tematyka ćwiczeń:</b> <ul style="list-style-type: none"> <li>• Zapoznanie z interfejsem środowiska programistycznego Microsoft Visual Studio.</li> <li>• Konwersja liczb pomiędzy systemami: dziesiętnym a dwójkowym, ósemkowym i heksadecymalnym.</li> <li>• Opracowywanie prostych algorytmów. Budowa schematów blokowych: wyznaczania bezwzględnej wartości liczby rzeczywistej, wyznaczania pierwiastków równania liniowego i kwadratowego, określenia, czy wprowadzona została liczba parzysta, wyznaczania średniej algebraicznej, wyznaczania silni.</li> <li>• Definiowanie struktury programu w języku C.</li> <li>• Definiowanie stałych i zmiennych w języku C, nadawanie zmiennym wartości początkowych.</li> <li>• Wyprowadzanie informacji na monitor z wykorzystaniem poznanych instrukcji: puts(), putchar(), printf() i cout.</li> <li>• Pobieranie danych z klawiatury z wykorzystaniem poznanych instrukcji: gets(), getchar(), scanf() i cin.</li> <li>• Integracja fazy pobierania danych, przetwarzania i wyprowadzania informacji.</li> <li>• Analiza przykładowych programów w języku C.</li> <li>• Wykorzystanie instrukcji warunkowych: if, if...else, if...else if...else.</li> <li>• Wykorzystanie rekurencji i pętli: for, while i do...while.</li> </ul>	
<b>Formy i kryteria zaliczenia przedmiotu/modułu</b>  Pisemne kolokwium wymagające znajomości zagadnień omawianych na ćwiczeniach. Projekt programistyczny – obejmuje wytworzenie oprogramowania służącego do rozwiązywania prostego zadania, np. z zakresu matematyki, fizyki. Zaliczenie wykładów – forma pisemna, obejmuje pytania teoretyczne oraz zadania z zakresu podstaw programowania.	Procentowy udział w końcowej ocenie  70% oceny z ćwiczeń 30% oceny z ćwiczeń 100% oceny z zaliczenia wykładów
<b>WYKAZ LITERATURY</b>	
<ul style="list-style-type: none"> <li>• Deitel H.M., Deitel P.J. (1998): Arkana C++ Programowanie. Wydawnictwo RM, Warszawa</li> <li>• Grębosz J. (2010): Symfonia C++ Standard. Wydanie trzecie B – rozszerzone i poprawione. Edition 2000, Kraków.</li> <li>• Krok E., Stempnakowski Z. (2008): Podstawy algorytmów. Schematy Blokowe. Difin, Warszawa</li> <li>• Neibauer A.R. (2004): Języki C i C++. Twój pierwszy program. Komputerowa Oficyna Wydawnicza Help, Michałowice.</li> <li>• Stroustrup Bjarne (2013). Programowanie. Teoria i praktyka z wykorzystaniem C++. Wydanie II poprawione. Helion, Gliwice.</li> </ul>	