



Programowanie w C# Karta opisu przedmiotu

Informacje podstawowe

Kierunek studiów Informatyka Społeczna	Cykl dydaktyczny 2026/2027	
Specjalność Wszystkie	Kod przedmiotu HIFSS.II1.07644.26	
Jednostka organizacyjna Wydział Humanistyczny	Języki wykładowe polski	
Poziom kształcenia Studia magisterskie II stopnia	Obligatoryjność Obowiązkowy	
Forma studiów Stacjonarne	Blok zajęciowy Przedmioty podstawowe	
Profil studiów Praktyczny	Przedmiot powiązany z badaniami naukowymi Tak	
Koordinator przedmiotu	Artur Chłosta	
Prowadzący zajęcia	Artur Chłosta	
Okres Semestr 1	Forma zaliczenia Zaliczenie	Liczba punktów ECTS 3
	Forma prowadzenia i godziny zajęć Ćwiczenia laboratoryjne: 30	

Efekty uczenia się dla przedmiotu

Kod	Efekty w zakresie	Kierunkowe efekty uczenia się	Metody weryfikacji
Wiedzy - Student zna i rozumie:			
W1	Student zna język programowania C#	IFS2P_W02	Wykonanie ćwiczeń laboratoryjnych, Zaliczenie laboratorium
Umiejętności - Student potrafi:			

Kod	Efekty w zakresie	Kierunkowe efekty uczenia się	Metody weryfikacji
U1	Student potrafi wykorzystywać język C# do rozwiązywania problemów praktycznych .	IFS2P_U05	Aktywność na zajęciach, Wykonanie ćwiczeń laboratoryjnych
U2	Student potrafi zbudować program w języku C#	IFS2P_U04	Aktywność na zajęciach, Wykonanie ćwiczeń laboratoryjnych
Kompetencji społecznych - Student jest gotów do:			
K1	Student ma świadomość znaczenia znajomości programowania przy rozwiązywaniu problemów praktycznych.	IFS2P_K02	Aktywność na zajęciach, Wykonanie ćwiczeń laboratoryjnych

Treści programowe zapewniające uzyskanie efektów uczenia się dla modułu zajęć

Celem ćwiczeń laboratoryjnych będzie zaznajomienie studentów z językiem programowania C# w stopniu umożliwiającym tworzenie prostych programów w tym języku.

Nakład pracy studenta

Rodzaje zajęć studenta	Średnia liczba godzin* przeznaczonych na zrealizowane aktywności
Ćwiczenia laboratoryjne	30
Przygotowanie do zajęć	25
Samodzielne studiowanie tematyki zajęć	20
Dodatkowe godziny kontaktowe	2
Łączny nakład pracy studenta	Liczba godzin 77
Liczba godzin kontaktowych	Liczba godzin 30

* godzina (lekcyjna) oznacza 45 minut

Treści programowe

Lp.	Treści programowe	Efekty uczenia się dla przedmiotu	Formy prowadzenia zajęć
1.	<p>Celem ćwiczeń laboratoryjnych będzie zaznajomienie studentów z językiem programowania C# w stopniu umożliwiającym tworzenie prostych programów w tym języku</p> <p>Tematyka zajęć:</p> <p>Wstęp do programowania w C# Komentarze, regiony Typy podstawowe Operatory Metody Tablice Pętle Instrukcje warunkowe Typy danych Typy generyczne Wnioskowanie typu (var) Typy wyliczeniowe Wstęp do obiektowości</p> <p>Klasy Struktury Konstruktory Pola Właściwości Zdarzenia Polimorfizm Delegaty Metody anonimowe Indeksery Interfejsy</p>	W1, U1, U2, K1	Ćwiczenia laboratoryjne

Informacje rozszerzone

Metody i techniki kształcenia :

Dyskusja

Rodzaj zajęć	Metody zaliczenia	Warunki zaliczenia przedmiotu
Ćwiczenia laboratoryjne	Aktywność na zajęciach, Wykonanie ćwiczeń laboratoryjnych, Zaliczenie laboratorium	

Warunki i sposób zaliczenia poszczególnych form zajęć, w tym zasady zaliczeń poprawkowych, a także warunki dopuszczenia do egzaminu

Warunkiem uzyskania zaliczenia jest wykonanie ćwiczeń. Sposób zaliczania nie zmienia się w kolejnych terminach.

Sposób obliczania oceny końcowej

100% oceny stanowił będzie wykonywanie ćwiczeń laboratoryjnych w formie zadań na zajęciach. Obecność na zajęciach i aktywny w nich udział są warunkiem koniecznym do uzyskania zaliczenia. W przypadku nieobecności student musi wykonać obowiązkowe ćwiczenia w czasie i formie uzgodnionej z prowadzącym. 2 i 3 termin będą odbywały się w sesji i sesji poprawkowej. Zaliczenia będzie polegał na wykonaniu zadanych ćwiczeń

Sposób i tryb wyrównywania zaległości powstałych wskutek nieobecności studenta na zajęciach

Zaległości student może nadrobić w oparciu o literaturę zaleconą przez wykładowcę. Powstałe zaległości student zalicza w terminie ustalonym z wykładowcą.

Wymagania wstępne i dodatkowe

Obecności są wymagane w ramach nieobecności student/ka musi w ramach dyżuru zaliczyć wymagane ćwiczenia i/lub literaturę.

Zasady udziału w poszczególnych zajęciach, ze wskazaniem, czy obecność studenta na zajęciach jest obowiązkowa

Ćwiczenia laboratoryjne: Studenci wykonują ćwiczenia laboratoryjne zgodnie z materiałami udostępnionymi przez prowadzącego. Student jest zobowiązany do przygotowania się w przedmiocie wykonywanego ćwiczenia, co może zostać zweryfikowane kolokwium w formie ustnej lub pisemnej. Zaliczenie zajęć odbywa się na podstawie zaprezentowania rozwiązania postawionego problemu. Zaliczenie modułu jest możliwe po zaliczeniu wszystkich zajęć laboratoryjnych.

Literatura

Obowiązkowa

1. C# 8.0. Kompletny przewodnik dla praktyków (Essential C# 8.0), Mark Michaelis, Eric Lippert
2. C# 9.0 w pigułce (C# 9.0 in a Nutshell), Joseph Albahari
3. C# in Depth, Jon Skeet
4. Dokumentacja języka C#, <https://msdn.microsoft.com/en-us/library/kx37x362.aspx>
5. Bardziej efektywny C#, Wagner Bill
6. Tajniki C# i .NET Framework. Wydajne aplikacje dzięki zaawansowanym funkcjom języka C# i architektury .NET, Posadas Marino
7. C#. Lekcje programowania. Praktyczna nauka programowania dla platform .NET i .NET Core, Jacek Matulewski
8. Programowanie gier przy użyciu Unity i C#, Casey Hardman

Dodatkowa

1. Unity i C# Podstawy programowania gier, Ewa Ross, Jacek Ross
2. Algorytmy, struktury danych i techniki programowania, Piotr Wróblewski

Badania i publikacje

Publikacje

1. Celem kursu jest zapoznanie uczestników z podstawowymi elementami składni języka programowania C#, w sposób umożliwiający wykorzystanie zdobytej wiedzy na kolejnych etapach kształcenia.
2. Ze względu na podstawowy charakter kursu nie przypisano publikacji prowadzącego do zajęć.

Kierunkowe efekty uczenia się

Kod	Treść
IFS2P_K02	ma świadomość znaczenia interdyscyplinarnej wiedzy z zakresu nauk społecznych i technicznych podczas identyfikacji i rozstrzygnięcia dylematów oraz związanych ze złożonymi relacjami technologii i świata społecznego przy rozwiązywaniu problemów poznawczych i praktycznych
IFS2P_U04	potrafi formułować i testować hipotezy związane z prostymi problemami wdrożeniowymi; umie zaprojektować zaawansowane rozwiązania techniczne (w formie urządzeń, obiektów, systemów lub procesów) oraz ma umiejętności pozwalające na ich realizację
IFS2P_U05	potrafi prowadząc debatę wykorzystać zaawansowaną wiedzę do szczegółowego opisu i praktycznej analizy procesów związanych z przemianami dokonującymi się w społeczeństwie pod wpływem technologii, potrafi również na zaawansowanym poziomie przetwarzać dane i wykorzystywać odpowiednie metody i narzędzia, w tym zaawansowane techniki informacyjno-komunikacyjne
IFS2P_W02	ma zaawansowaną wiedzę z zakresu metod i technik programowania, konstrukcji i ekstrakcji baz danych oraz grafiki komputerowej oraz projektowania interfejsów, interakcji i doświadczeń użytkownika