

Karta opisu zajęć - Sylabus			
Państwowa Akademia Nauk Stosowanych im. ks. Bronisława Markiewicza w Jarosławiu			
I. INFORMACJE PODSTAWOWE			
Nazwa zajęć: Programowanie obiektowe		Cykl kształcenia rozpoczynający się w roku akademickim 2024/2025	
Nazwa kierunku studiów, poziom i profil kształcenia: Informatyka, studia pierwszego stopnia, profil praktyczny			
Język wykładowy: polski/angielski		Rodzaj zajęć: zajęcia kształcenia kierunkowego	
Rok studiów: 2		Semestr: 3	
Liczba punktów ECTS przypisana zajęciom: 6		Koordynator zajęć Imię, nazwisko, tytuł/stopień naukowy, adres e-mail:	
Jednostka organizacyjna: Wydział Inżynierii Technicznej			
FORMA PROWADZENIA ZAJĘĆ I LICZBA GODZIN			
Ogólna liczba godzin zajęć dydaktycznych na studiach stacjonarnych i niestacjonarnych z podziałem na formy:			
Studia stacjonarne		Studia niestacjonarne	
Wykład:	30	Wykład:	
Ćwiczenia:		Ćwiczenia:	
Laboratorium:	45	Laboratorium:	
Lektorat:		Lektorat:	
Projekt:		Projekt:	
Zajęcia praktyczne:		Zajęcia praktyczne:	
Seminarium:		Seminarium:	
Zajęcia terenowe:		Zajęcia terenowe:	
Praktyki zawodowe:		Praktyki zawodowe:	
Inna forma (jaka):		Inna forma (jaka):	
RAZEM:	75	RAZEM:	
II. INFORMACJE SZCZEGÓŁOWE			

Wymagania wstępne i dodatkowe:

wymagania formalne - student powinien być zarejestrowany na semestr trzeci

wymagania wstępne - w zakresie Wiedzy: Student powinien posiadać wiedzę w zakresie realizowanym w ramach przedmiotu Algorytmy i struktury danych, Podstawy programowania.

w zakresie Umiejętności: student powinien posiadać podstawowe umiejętności w zakresie posługiwania się systemem komputerowym wyniesione z kształcenia na roku pierwszym w ramach przedmiotów: Algorytmy i struktury danych, Podstawy programowania,

wymagania w zakresie Kompetencji społecznych: Zrozumienie potrzeby ciągłego dokształcania się, umiejętność pozyskiwania informacji, umiejętność samokształcenia się.

Cel (cele) kształcenia dla zajęć:

Celem kształcenia jest nabycie przez studentów wiedzy i umiejętności z zakresu programowania obiektowego, zapoznanie z metodologią i zasadami programowania obiektowego oraz współczesnym środowiskiem języka programowania obiektowego Microsoft Visual Studio.

EFEKTY UCZENIA SIĘ OKREŚLONE DLA ZAJĘĆ I ICH ODNIESIENIE DO EFEKTÓW UCZENIA SIĘ OKREŚLONYCH DLA KIERUNKU STUDIÓW

Efekty uczenia się określone dla zajęć w kategorii wiedza, umiejętności oraz kompetencje społeczne oraz metody weryfikacji efektów uczenia się

UWAGA:

Dzielimy efekty uczenia się określone dla zajęć na kategorie wiedzy, umiejętności i kompetencji społecznych. Określone dla zajęć efekty uczenia się nie muszą obejmować wszystkich trzech kategorii i zależą one od formy zajęć.

Symbol efektów uczenia się określonego dla zajęć*	Treść efektu uczenia się. Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się, student w kategorii:	Odniesienie do efektów uczenia się określonych dla kierunku studiów (symbol efektów uczenia się)
Wiedzy - zna i rozumie		
E_01	zna składnię języka, struktury danych oraz potrafi objaśnić działanie instrukcji w wybranym języku programowania obiektowego	K_W08
E_02	zna i rozumie zastosowanie metodologii programowania obiektowego podczas rozwiązywania problemów informatycznych	K_W08
E_03	potrafi sformułować prosty algorytm oraz napisać program komputerowy rozwiązujący typowe zagadnienia obliczeniowe wykorzystując techniki programowania obiektowego	K_W08
Umiejętności - potrafi		
E_04	Wykonać projekt aplikacji zawierającej schemat klasy. Potrafi dodawać składniki klas w postaci pól i metod.	K_U08, K_U14
E_05	Budować schematy klas wykorzystujących podstawowe zagadnienia programowania obiektowego jak abstrakcja, hermetyzacja, polimorfizm, dziedziczenie.	K_U08, K_U14
E_06	Wykonać aplikacje okienkowe z graficznym interfejsem użytkownika korzystające z paradygmatów programowania obiektowego	K_U12, K_U14

Kompetencji społecznych - jest gotów do				
E_07	potrafi działać w grupie, ma świadomość odpowiedzialności za pracę własną i całego zespołu, rozumie potrzebę systematycznej pracy w celu zdobywania wyższych kompetencji zawodowych.		K_K03, K_K05	
UWAGA! Zaleca się, aby w zależności od liczby godzin zajęć, liczba efektów uczenia się zawierała się w przedziale: 3-7, ale są to wartości umowne.				
TREŚCI PROGRAMOWE I ICH ODNIESIENIE DO FORM ZAJĘĆ I METOD OCENIANIA				
Treści programowe (uszczegółowione, zaprezentowane z podziałem na poszczególne formy zajęć, tj. wykład, ćwiczenia, laboratoria, projekty, seminaria i inne):				
Symbol treści programowych	Opis treści programowych	Forma zajęć	Metody dydaktyczne prowadzenia zajęć umożliwiające osiągnięcie założonych efektów uczenia się *	Metody weryfikacji osiągnięcia efektów uczenia się przypisanych do zajęć #
		wykład		
TP-01	Omówienie paradygmatu programowania obiektowego – definiowanie programów za pomocą obiektów. Podstawowe założenia paradygmatu obiektowego: Abstrakcja, Hermetyzacja, Polimorfizm, Dziedziczenie.		Wykład problemowy - konwersacyjny	Egzamin pisemny
TP-02	Przedstawienie zasad programowania zorientowanego obiektowo w C++. Zasady definiowania klas oraz tworzenia obiektów Omówienie przykładów programów./ Characteristics of object-oriented programming (classes and their components, objects,). Examples of programs.		Wykład problemowy - konwersacyjny	Egzamin pisemny
TP-03	Zasady Hermetyzacji obiektów w C++ . Zadania konstruktorów oraz destruktora. Omówienie przykładów programów.		Wykład problemowy - konwersacyjny	Egzamin pisemny

TP-04	Rola funkcji oraz klas zaprzyjaźnionych z innymi klasami. Przeładowanie operatorów. Zadania konstruktora kopiującego oraz operatora przypisania.		Wykład problemowy - konwersacyjny	Egzamin pisemny
TP-05	Omówienie tematu wykorzystania klas na zasadzie Dziedziczenia. Pojęcie klasy bazowej i klasy pochodnej. Omówienie przykładów programów.		Wykład problemowy - konwersacyjny	Egzamin pisemny
TP-06	Hierarchie dziedziczenia. Zalety i wady dziedziczenia wielokrotnego. Omówienie mechanizmu dziedziczenia w innych językach obiektowych: C#, Java. Omówienie możliwości oraz wykorzystania funkcji wirtualnych. Omówienie przykładów programów.		Wykład problemowy - konwersacyjny	Egzamin pisemny
TP-07	Przedstawienie zastosowania klas abstrakcyjnych w językach C++, C#. Omówienie mechanizmu obsługi wyjątków w C++.		Wykład problemowy - konwersacyjny	Egzamin pisemny
TP-08	Podstawowe informacje na temat platformy programistycznej .NET Tworzenie graficznych interfejsów użytkownika – aplikacje w C#. Komponenty interfejsu graficznego użytkownika. Zasady obsługi zdarzeń w aplikacjach Windows.		Wykład problemowy - konwersacyjny	Egzamin pisemny
		laboratorium		
TP-09	Tworzenie nowych projektów w środowisku programistycznym Microsoft Visual Studio. Podział kodu źródłowego na wiele plików. Wstępne przygotowanie kodu źródłowego poszczególnych klas. Tworzenie nowych obiektów - testowanie aplikacji.		ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy, studium przypadku	Zaliczenie ustne, projekt

TP-10	Rozbudowa aplikacji poprzez dodanie nowych metod. Pisanie definicji funkcji do których obiekty przekazywane są w formie argumentu. Definiowanie Konstruktor i Destruktora w przykładowych klasach. Uruchamianie i testowanie aplikacji		ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy, studium przypadku	Zaliczenie ustne, projekt
TP-11	Rozbudowa aplikacji poprzez dodanie w schemacie klasy mechanizmu funkcji zaprzyjaźnionej z klasą – testowanie aplikacji. Definiowanie konstruktora kopiującego i przedefiniowanie operatora przypisania – testowanie aplikacji.		ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy, studium przypadku	Zaliczenie ustne, projekt
TP-12	Rozbudowa aplikacji poprzez dodanie nowych klas dziedziczących. Modyfikacja aplikacji poprzez zdefiniowanie w kodzie funkcji wirtualnej oraz destruktora wirtualnego – testowanie aplikacji.		ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy, studium przypadku	Zaliczenie ustne, projekt
TP-13	Tworzenie aplikacji Windows Forms – tworzenie prostych interfejsów użytkownika w C++ i C# – testowanie aplikacji.		ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy, studium przypadku	Zaliczenie ustne, projekt
TP-14	Tworzenie aplikacji okienkowej Windows Forms wykorzystującej schematy klas z poprzednich zajęć. Tworzenie interfejsów użytkownika oraz pisanie kodów źródłowych obsługi zdarzeń.		ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy, studium przypadku	Zaliczenie ustne, projekt
TP-15	Realizacja aplikacji Windows Forms w języku C# operującej na danych w postaci obiektów. Aplikacja wykorzystuje możliwości zapisu i odczytu informacji o obiektach w plikach tekstowych – prosta baza danych.		ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy, studium przypadku	Zaliczenie ustne, projekt

Metody weryfikacji osiągnięcia efektów uczenia się określonych dla zajęć, powinny być zróżnicowane w zależności od kategorii, tj. inne dla kategorii wiedza i inne dla kategorii umiejętności i kompetencje społeczne.

Dla wykładu:

* np. wykład podający, wykład problemowy, ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy

np. egzamin ustny, test, prezentacja, projekt

Zaleca się podanie przykładowych zadań (pytań) służących weryfikacji osiągnięcia efektów uczenia się określonych dla zajęć.

ZAŁECANA LITERATURA (w tym pozycje w języku obcym)

Literatura podstawowa (powinna być dostępna dla studenta w uczelnianej bibliotece):

1. Grębosz J.: „Symfonia C++ standard”, Edycja 2008
2. Eckel B.: „Thinking in C++”, Helion, Gliwice, 2006,
3. Matulewski J.: „Visual Studio 2010 dla programistów c#”, Helion, Gliwice, 2011,
4. Griffiths I.: „C# 8.0. Programowanie. Tworzenie aplikacji Windows, internetowych oraz biurowych”, Helion, Gliwice, 2020,

Literatura uzupełniająca:

1. Snell M., Power L. ; [tł. Tomasz Walczak]: „Microsoft Visual Studio 2010 : księga eksperta”, Helion 2011,
2. Boduch A.: „Wstęp do programowania w C#, Helion”, Gliwice, 2006
3. Troelsen S.: „Język C# i Platforma .NET”, Wydawnictwo Naukowe PWN, Warszawa 2006
4. Notatki z wykładu

III. INFORMACJE DODATKOWE

BILANS PUNKTÓW ECTS

OBCIĄŻENIE PRACĄ STUDENTA (godziny)

Forma aktywności	Liczba godzin *
Godziny zajęć (według harmonogramu) z nauczycielem akademickim lub inną osobą prowadzącą zajęcia	75
Praca własna studenta	75
SUMA GODZIN:	150

OBCIĄŻENIE PRACĄ STUDENTA (punkty ECTS)

		Liczba punktów ECTS	
SUMARYCZNA LICZBA PUNKTÓW ECTS PRZYPISANYCH DO ZAJĘĆ	Praca studenta wymagająca bezpośredniego kontaktu z nauczycielem akademickim lub inną osobą prowadzącą zajęcia	Ogółem: 6	3
	Praca własna studenta		3

* godziny lekcyjne, czyli 1 godz. oznacza 45 min;

OPIS PRACY WŁASNEJ STUDENTA:

Praca własna studenta musi być precyzyjnie opisana, uwzględniając charakter praktyczny zajęć. Należy podać symbol efektu uczenia się, którego praca własna dotyczy oraz metody weryfikacji efektów uczenia się stosowane w ramach pracy własnej.

Przykładowe formy aktywności: (1) przygotowanie do zajęć, (2) opracowanie wyników, (3) czytanie wskazanej literatury, (4) napisanie raportu z zajęć, (5) przygotowanie do egzaminu, opracowanie projektu.

Samodzielna praca związana z czytaniem wskazanej literatury oraz przygotowanie do egzaminu (E_01 – E_03).

Samodzielna realizacja ćwiczeń programistycznych na podstawie instrukcji, przygotowanie mini-projektów indywidualnych oraz grupowych (E_04 – E_07).

KRYTERIA OCENIANIA

Zajęcia laboratoryjne kończą się zaliczeniem na ocenę, zaś wykład kończy się egzaminem.

Ocena kształtująca:

Forma i warunki zaliczenia wykładu:

- obecność na wykładach zgodna z Regulaminem Studiów
- Egzamin pisemny
- uzyskanie z egzaminu oceny pozytywnej – co najmniej dostatecznej (min. 51% punktów poprawnych odpowiedzi)

Forma i warunki zaliczenia laboratorium:

- zaliczenie na ocenę
- frekwencja (zgodnie z Regulaminem Studiów)
- zaliczenia ustne z wykonywanych ćwiczeń projektowych (programów) wykonywanych podczas zajęć
- ocena końcowa jest średnią ocen z zaliczeń cząstkowych lub z ewentualnego wykonywanego projektu zaliczeniowego

Ocena podsumowująca:

Na ocenę dostateczną student zrealizował projekty programistyczne dotyczące opisywanych treści programowych przedmiotu w zakresie podstawowym i potrafi omówić zastosowane techniki. Ponadto w dostatecznym stopniu potrafi wykorzystać przekazaną wiedzę podczas wykładu w realizowanych projektach programistycznych oraz podczas egzaminu pisemnego.

Na ocenę dobrą student zrealizował kompletne projekty programistyczne dotyczące opisywanych treści programowych przedmiotu i potrafi omówić zastosowane techniki oraz uzasadnić ich wybór. Ponadto w dobrym stopniu potrafi wykorzystać przekazaną wiedzę podczas wykładu w realizowanych projektach programistycznych oraz podczas egzaminu pisemnego.

Na ocenę bardzo dobrą student zrealizował kompletne projekty programistyczne dotyczące opisywanych treści programowych przedmiotu z zastosowaniem ponadstandardowych rozwiązań i potrafi omówić zastosowane techniki oraz uzasadnić ich wybór. Ponadto w bardzo dobrym stopniu potrafi wykorzystać przekazaną wiedzę podczas wykładu w realizowanych projektach programistycznych oraz podczas egzaminu pisemnego.

INFORMACJA O PRZEWIDYWANEJ MOŻLIWOŚCI WYKORZYSTANIA KSZTAŁCENIA NA ODLEGŁOŚĆ

Istnieje możliwość prowadzenia części zajęć (wykład) z wykorzystaniem metod i technik kształcenia na odległość z wykorzystaniem platformy Microsoft Teams zapewniającej synchroniczny, interaktywny sposób komunikowania się, jako formy uzupełniającej kształcenie prowadzone w siedzibie Uczelni.