

Karta opisu zajęć - Sylabus

Państwowa Wyższa Szkoła Techniczno-Ekonomiczna im. ks. Bronisława Markiewicza w Jarosławiu

I. INFORMACJE PODSTAWOWE

Nazwa zajęć: Języki programowania wysokiego poziomu	Cykl kształcenia: 2022-2023	Data aktualizacji sylabusu: 15.02.2023
Nazwa kierunku studiów, poziom i profil kształcenia: Automatyka i Elektronika Praktyczna, pierwszego stopnia, praktyczny		
Język wykładowy: polski	Rodzaj zajęć: Zajęcia kształcenia kierunkowego	
Rok studiów: I	Semestr: II	
Liczba punktów ECTS przypisana zajęciom: 3	Koordynator zajęć Monika Piróg, dr, monika.pirog@pwste.edu.pl	
Jednostka organizacyjna: Instytut Inżynierii Technicznej	Prowadzący zajęcia Monika Piróg, dr, monika.pirog@pwste.edu.pl	

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:	15	Wykład:	
Ćwiczenia:		Ćwiczenia:	
Laboratorium:	45	Laboratorium:	
Lektorat:		Lektorat:	
Projekt:		Projekt:	
Zajęcia praktyczne:		Zajęcia praktyczne:	
Seminarium:		Seminarium:	
Zajęcia terenowe:		Zajęcia terenowe:	
Praktyki:		Praktyki:	
Inna forma (jaka):		Inna forma (jaka):	
RAZEM:	60	RAZEM:	

II. INFORMACJE SZCZEGÓŁOWE

Wymagania wstępne i dodatkowe:

Wymagania wstępne w zakresie wiedzy, umiejętności oraz kompetencji społecznych (jeśli obowiązują):
wymagania formalne - student powinien być zarejestrowany na semestr drugi
wymagania wstępne – Podstawy programowania – znajomość przedmiotu w stopniu dobrym.

Cel (cele) kształcenia dla zajęć:	
Celem przedmiotu jest omówienie współczesnych technik programowania, w tym programowania strukturalnego, obiektowego, obiektowo-orientowanego i funkcjonalnego na bazie wybranych języków programowania (C++, C# i Java).	
Efekty uczenia się określone dla zajęć	
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 od formy zajęć.	
Symbol efektów uczenia się określonego dla zajęć*	Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się, student w kategorii:
Wiedzy - zna i rozumie	
M_01	Zna składnię i semantykę języków programowania, sposoby przekazywania parametrów do podprogramów, potrafi zdefiniować abstrakcyjne typy danych oraz wyjaśnić zasady przeciążania operatorów.
M_02	Zna i rozumie zastosowanie metodologii programowania obiektowego podczas rozwiązywania problemów informatycznych.
M_03	Ma uporządkowaną wiedzę z zakresu języków programowania wysokiego poziomu, zna zasady doboru języka programowania do rozwiązywania problemów w zakresie oprogramowania sprzętu i usług; rozumie metody specyfikowania podstawowych wymagań w zakresie oprogramowania.
Umiejętności - potrafi	
M_04	Potrafi pracować indywidualnie i w zespole; umie oszacować czas potrzebny na realizację zleconego zadania; potrafi opracować i zrealizować harmonogram prac zapewniający dotrzymanie terminów.
M_05	Ma umiejętność tworzenia projektów programistycznych w oparciu o języki programowania wysokiego poziomu.
M_06	Potrafi korzystać z dokumentacji i specyfikacji technicznych w celu dobrania odpowiednich parametrów i komponentów projektowanego systemu.
M_07	Potrafi sformułować algorytm, posługuje się językami programowania wysokiego poziomu oraz odpowiednimi narzędziami informatycznymi do opracowania programów komputerowych.
Kompetencji społecznych - jest gotów do	
M_08	Rozumie potrzebę uczenia się przez całe życie.
M_09	Służy wsparciem i pomocą członkom zespołu w trudnych i stresujących sytuacjach.
TREŚCI PROGRAMOWE I ICH ODNIESIENIE DO EFEKTÓW UCZENIA SIĘ OKREŚLONYCH DLA ZAJĘĆ	
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	Liczba godzin	Odniesienie do efektów uczenia się określonych dla zajęć
		wykład		
TP-01	Omówienie tematyki przedmiotu, literatury, form i zasad zaliczenia. Pojęcia ogólne: opis składni i semantyki języków programowania – C++, C# i Java. Omówienie aktualnych trendów rozwojowych w poszczególnych językach programowania.		2	M_01,
TP-02	Wykorzystanie abstrakcyjnych typów danych. Możliwości wykorzystania przeciążania operatorów dla własnych typów danych. Przekazywanie argumentów do funkcji w poszczególnych językach programowania.		2	M_01, M_02,
TP-03	Mechanizmy zarządzania pamięcią w języku C++, C# oraz Java. Przykłady programów. Omówienie cech charakterystycznych programowania imperatywnego.		2	M_01, M_03,
TP-04	Przedstawienie cech charakterystycznych programowania obiektowego dla poszczególnych języków C++, C# i Java. Definiowanie klas: prawa dostępu, konstruktor, destruktor, lista inicjacyjna, pola i funkcje statyczne, funkcje zaprzyjaźnione, tworzenie obiektów, dostęp do obiektów. Omówienie przykładowych programów z języków: C++, Java, C#.		2	M_01, M_02, M_03,
TP-05	Omówienie możliwości dziedziczenia i polimorfizmu w językach obiektowych. Pojęcie klasy bazowej i pochodnej, przesłanianie składowych, wiązanie statyczne i dynamiczne, funkcje wirtualne, klasy abstrakcyjne, interfejsy.		2	M_01, M_02, M_3,

TP-06	Przykłady projektów realizujących zadania programistyczne wykorzystujących możliwości programowania obiektowego. Omówienie przykładów w wybranych językach C# i Java.		2	M_01, M_02, M_03, M_08,
TP-07	Programowanie graficznych interfejsów użytkownika. Przegląd komponentów oraz ich właściwości. Okna i rozmieszczenie komponentów. Zalety i wady ręcznego programowania interfejsu użytkownika.		2	M_01, M_02, M_03, M_08,
TP-08	Możliwości szybkiego tworzenia graficznych interfejsów użytkownika w środowisku Eclipse, Microsoft Visual Studio.		1	M_01, M_02, M_03, M_08,
		laboratorium		
TP-09	Zajęcia praktyczne – zapoznanie ze środowiskiem programistycznym Microsoft Visual Studio oraz Eclipse. Tworzenie nowych projektów. Pisanie kodów źródłowych w celu powtórzenia podstawowych typów danych, funkcji, pętli, tablic.		4	M_01, M_02, M_03, M_04, M_05, M_06, M_07, M_08, M_09
TP-10	Tworzenie i usuwanie tablic dynamicznych w języku C++. Pisanie programów oraz testowanie mechanizmu automatycznego zarządzania pamięcią w językach C# i Java. Tworzenie przykładowych programów wykorzystujących obiekty. Tworzenie schematów klas, konstruktorów, destruktorów. Pisanie funkcji składowych. Przekazywanie obiektów do funkcji w postaci argumentu. Funkcje zaprzyjaźnione oraz ich możliwości. Testowanie możliwości dostępu do składników obiektu, uruchamianie funkcji prywatnych. Przykłady w języku C i Java.		10	M_01, M_02, M_03, M_04, M_05, M_06, M_07, M_08, M_09
TP-11	Zajęcia praktyczne - tworzenie klas dziedziczących z wcześniej utworzonych, tworzenie klas abstrakcyjnych i interfejsów oraz ich wykorzystanie w nowych klasach – pisanie przykładowych programów w języku C i Java. Dostęp do składników obiektów złożonych. Pisanie oraz testowanie funkcji wirtualnych.		6	M_01, M_02, M_03, M_04, M_05, M_06, M_07, M_08, M_09

TP-12	Tworzenie aplikacji Windows Forms – tworzenie prostych interfejsów użytkownika w C++ i C# – testowanie aplikacji. Techniki zarządzania układem graficznym. Układ komponentów. Obsługa zdarzeń. Komponenty – wprowadzania tekstu, dokonywania wyboru. Okna dialogowe. Budowa menu.		8	M_01, M_02, M_03, M_04, M_05, M_06, M_07, M_08, M_09
TP-13	Tworzenie prostych aplikacji i apletów w języku Java. Układ graficzny aplikacji. Obsługa zdarzeń – interfejs nasłuchu zdarzeń. Pola wyboru, okna dialogowe.		8	M_01, M_02, M_03, M_04, M_05, M_06, M_07, M_08, M_09
TP-14	Realizacja aplikacji okienkowej w języku C# lub Java operującej na danych w postaci obiektów. Aplikacja wykorzystuje możliwości zapisu i odczytu informacji o obiektach w plikach tekstowych – prosta baza danych.		9	M_01, M_02, M_03, M_04, M_05, M_06, M_07, M_08, M_09
ZALECANA LITERATURA (w tym pozycje w języku obcym)				
Literatura podstawowa (powinna być dostępna dla studenta w uczelnianej bibliotece):				
<ol style="list-style-type: none"> Grębosz J.: Symfonia C++ standard, Edycja 2015 Snell M., Power L.: „Microsoft Visual Studio 2010, Helion, Gliwice, 2011, Schildt H.: „Java : sztuka programowania”, Helion, Gliwice, 2004 				
Literatura uzupełniająca:				
<ol style="list-style-type: none"> Boduch A.: Wstęp do programowania w C#, Helion, Gliwice, 2006 Troelsen S.: Język C# i Platforma .NET, Wydawnictwo Naukowe PWN, Warszawa2006 				
III. INFORMACJE DODATKOWE				
Odniesienie efektów uczenia się określonych dla zajęć i treści programowych do form zajęć i metod oceniania				
Symbol efektu uczenia się określonego dla zajęć	Symbol treści programowych realizowanych w trakcie zajęć	Formy zajęć i metody dydaktyczne prowadzenia zajęć umożliwiające osiągnięcie założonych efektów uczenia się *	Metody weryfikacji osiągnięcia efektów uczenia się przypisanych do zajęć #	
Wiedza				
M_01	TP_01 – TP_14	Wykład i zajęcia praktyczne – prezentacja, wykład z dyskusją, praktyczna realizacja aplikacji	Test, kolokwium, projekt zaliczeniowy	
M_02	TP_02 – TP_14	Wykład i zajęcia praktyczne – prezentacja, wykład z dyskusją, praktyczna realizacja aplikacji	Test, kolokwium, projekt zaliczeniowy	
M_03	TP_04 – TP_14	Wykład i zajęcia praktyczne – prezentacja, wykład z dyskusją, praktyczna realizacja aplikacji	Test, kolokwium, projekt zaliczeniowy	
Umiejętności				

M_04	M_04	TP_09 – TP_14	Wykład i zajęcia praktyczne – prezentacja, wykład z dyskusją, praktyczna realizacja aplikacji
M_05	M_05	TP_09 – TP_14	Wykład i zajęcia praktyczne – prezentacja, wykład z dyskusją, praktyczna realizacja aplikacji
M_06	M_06	TP_09 – TP_14	Wykład i zajęcia praktyczne – prezentacja, wykład z dyskusją, praktyczna realizacja aplikacji
M_07	M_07	TP_09 – TP_14	Wykład i zajęcia praktyczne – prezentacja, wykład z dyskusją, praktyczna realizacja aplikacji
Kompetencje społeczne			
M_08	TP_06 – TP_14	Wykład i zajęcia praktyczne – prezentacja, wykład z dyskusją, praktyczna realizacja aplikacji	Test, kolokwium, projekt zaliczeniowy
M_09	TP_09 – TP_14	Wykład i zajęcia praktyczne – prezentacja, wykład z dyskusją, praktyczna realizacja aplikacji	Test, kolokwium, projekt zaliczeniowy

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.

BILANS PUNKTÓW ECTS

OBCIĄŻENIE PRACĄ STUDENTA (godziny)

Forma aktywności	Liczba godzin *
Godziny zajęć (według harmonogramu) z nauczycielem akademickim lub inną osobą prowadzącą zajęcia	60
Praca własna studenta	90
SUMA GODZIN:	150

OBCIĄŻENIE PRACĄ STUDENTA (punkty ECTS)

		Liczba punktów ECTS	
SUMARYCZNA LICZBA PUNKTÓW ECTS PRZYPIŚNANYCH DO ZAJĘĆ	Praca studenta wymagająca bezpośredniego kontaktu z nauczycielem akademickim lub inną osobą prowadzącą zajęcia	Ogółem: 5	2
	Praca własna studenta		3

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

OPIS PRACY WŁASNJE 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.

- przygotowanie do zajęć,
- przygotowanie projektu,
- czytanie wskazanej literatury

KRYTERIA OCENIANIA

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 zaliczenia 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 zaliczenia 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 zaliczenia pisemnego.

Podpis nauczyciela akademickiego lub osoby odpowiedzialnej za przedmiot:

.....
(imię i nazwisko)

.....
(podpis i data)

Podpis kierownika zakładu:

.....
(imię i nazwisko)

.....
(podpis i data)

Podpis dyrektora instytutu:

.....
(imię i nazwisko)

.....
(podpis i data)